PERSONALNI

računari

73

JUN 1991.
CENA 100 D

116 STRANA

NOVA SERIJA
BUDITE WINDOWS PROGRAMERI

RAČUNARI
GAMA i386/40 MHz
MOST STAR 486

KARTICE
HYPERSTORE 1600
SOUND BLASTER

UPOREDNI TESTOVI
ETHERNET KARTICE
TRI ARHIVERA

CLIPPER 5.01
BORLAND C++
SUPERBASE 4
OBJECTVISION
VIEWS DESIGNER
NOVELL 386

PROGRAMSKI JEZICI
POSTSCRIPT



YU ISSN 0352-7271

IBIS SYS
ekonomski biro

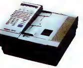
u saradnji sa

predstavlja

OKI
Oki Electric Industry Co., Ltd.

profesionalne telefaks aparate, vrhunske klase, OKIFAX

OKIFAX OF — 17H sa memorijom



Specifikacije:

* stoni tip CCITT G2/G3
* brzina prenosa 10 sec, A4
* valičina mamorije 30×A4
* veličina dokumanta A3 do A5
* korekcija greške CCITT, ECM
* dužina rolna 100 m

Funkcija:

* automstako ulaganja dokumenata do 40 stranica
* akraćeno biranje sa:
  - jednim tastarom 30 lokacija
  - dva cifre 70 lokacija
* alternativno biranje sa jednim tasterom 8 lokacija
* 16 nivoa sivog
* izabrana grupa korisnika 100 lokacija
* identifikacija korianika 16 alfanumeričkih karaktare
* posabna operacije:
  - difuzna otprema, odložena difuzija, relajna difuzija, odložena relejna difuzija, povarljiva otprama, povarljiva difuzija.
  - prijem preuzimanjem, otprama preuzimanjem, višestruko preuzimanje, odložano višestruko preuzimanje, inverzno prauzimanja.

Cena: 61.800,-

OKIFAX OF — 8M sa memorijom



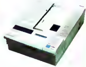
Tahnička karaktaristika:

* atoni tip CCITT, G3
* brzina pranoaa 13 sec, A4
* veličina memorija 7×A4
* prijem bez papira
* odloženo alanje
* odiožana difuzija
* prijem preuzimanjam
* 32 nivoa aivog
* korekcija greška, ECM

Cana: 38.800,-

Za sva ostale informacije
sa možete obratiti
Službi prodaje
biroopraema
na telefone:
609-650, 674-831,
606-193, 671-037
i telefax 674-741.

OKIFAX OF — 38 sa memorijom



Specifikacije:

* stoni tip CCITT G2/G3
* brzina prenoaa 10 sec A4
* veličina memorija 64×A4
* valičina dokumenta A3 do A5
* korekcija greške CCITT, ECM
* dužina rolna 100 m

Funkcija:

* automatsko uiaganja do 40 atranica
* akraćeno biranje sa:
  - jednim tastarom 60 lokacija
  - tri cifra 252 lokacija
* zatvorena grupa korianika 252 registrovanih lokacija
* 16 nivoa sivog
* povarljiva otprema i prijem sa 64 MAIL BOKSOVA
* posebna operacija:
  - difuzna otprema, odiožena difuzija, relajna difuzija, odložana ralejna difuzija, poverljiva otprema, poverljiva difuzija
  - prijem preuzimanjam, otprama prauzimanjam, odioženo višestruko preuzimanje, nivovsko prauzimanja
* izgradnja i uapostavljanje mreže sa posebnim funkcijama.

Cena: 99.900,-

PERSONALNI
računari 73

časopis za informatiku i računarstvo
YU ISSN 0352-7221
Izdaje BIGZ
jun 1991 cena 100 din

Izdaje i štampa
Beogradski izdavačko-grafički zavod
11000 Beograd
Bulevar Vojvode Mišića 17

Generalni direktor
Ilija Rapajić

Direktor Novinskog sektora
Zoran Nikodijević

Glavni i odgovorni urednik
Jovan Regasek

Zamenik gl. i odg. urednika
Esad Jakupović

Urednik za hardver
Dejan V. Veselinović

Tehnički urednik / sekretar redakcije
Natalija Desančić

Stručna redakcija
Vesna Ćosić (aktuelnosti), Nenad Batočanin (baze podataka), Miodrag Lović (radne tabele), Dejan Ristanović (softver), Jovan Skujlan (programiranje), Dejan Veselinović (hardver), Dragutin Vuković (mreže), Zoran Životić (softver)

Adresa redakcije
11000 Beograd
Bulevar vojvode Mišića 17/III

Telefoni
653 748 (redakcija)
650 528 (prodaja)

Agencija BIGZ-a
(011) 651-793, 653-565

Teleks
11855 BIGZ YU

Telefaks
(011) 648-140

Pretplata za zemlju
Za jednu godinu 1200
Za šest meseci 600
(na žiro račun: RO BIGZ, 60802-603-23264)

Pretplata za inostranstvo
Za jednu godinu 51 USD, 80 DEM, 67 CHS, 27 GBP, 271 FRF
(na devizni račun RO BIGZ 60811-620-16101-620701-999-03377)
Zbog usporenog prenošenja uplata preko banke molimo pretplatnike da nam pošle svake nove uplate odmah pošalju foto-kopiju uplatnice

Rukopisi se ne vraćaju

SEZAM
sistem za modemske komunikacije
(011) 648-693 24 časa dnevno
Upravnik sistema
Jovan Regasek
Administratori na sistemu
Zoran Životić i Dejan Ristanović
Moderatori
Jovan Bulajić (datoteke), Aleksandar Damnjanović (PC SOFT), Slobodan Kelezić (ORKA), Vladimir Kostić (PC SOFT), Vladimir Krstonošić (VICE VI), Esad Jakupović (RAČUNARI), Dalibor Lanik (ATARI ST), Dejan Ristanović (NOVOSTI I KOMUNIKACIJE), Bojan Tepavčević (FORUM), Dejan Veselinović (PC HARD) i Zoran Životić (SEZAM)
Sekretarica sistema
Natalija Desančić
Softver
Zoran Životić (glavna aplikacija) i Dejan Ristanović (eksterni programi)
Hardver
Server: COMPAQ DESKPRO 386/25, 3 MB RAM-a, tvrdi disk 300 MB (COMPAQ Minhen). Mreža: dve stanice COMPAQ 286N/12, 1 MB RAM-a (COMPAQ Minhen), tri stanice „Ultra" AT 12 MHz, D-LINK DE 100 LanSmart („Ultra" Skopje). Modemi ViVa (Thelema Communications, Beograd)

SADRŽAJ

4 Računari / Gama i386/40
U PETOJ BRZINI

6 Računari / Most Star 486
ZVEZDA PONUDE

8 Računari / Lightline notebook
SAVRŠEN PAR

10 Kontroleri / Hyperstore
FORMULA MEĐU DISKOVIMA

13 Rezači/skeneri / STRIKA
SAMOLEPLJIVI NATPISI

14 Zvučne kartice / Sound Blaster
SVIRKATA NA PISIITE

18 Seminari / Nove arhitekture
BRIGE NIKAD DOSTA

18 Kompajleri / Borland C++
C I VIŠE OD C-A

20 Kompajleri / Clipper 5.01
KRALJ JE STIGAO

26 Baze podataka / Superbase 4
DOSTOJAN SVOG IMENA

30 Razvojni alati / ObjectVision
GLEDAM I PROGRAMIRAM

32 Softverske alatke / Arhiveri
TRI ARHIVERA

34 Domaći softver
PRIVATNI STOMATOLOG

35 C biblioteke / Views Designer
DOBRO DOŠLI U VERMONT

36 CAD/CAM
OBJEKT I KAKO GA PREDSTAVITI

39 Programski jezici / PostScript
IZ MITA U LEGENDU

42 Operativni sistemi / Netware 386
NOVEL UZVRAĆA UDARAC

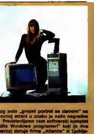
44 Mrežne kartice / Ethernet
NIKO KAO ETERNET

46 Lokalne mreže / Virusi
DEZIFINCIRAJTE MREŽU!

48 Komunikacije / Sezam
KAD BI SE VEZALI NA JUPAK

50 Dejan Ristanović
SEZAM BILTEN

51 Tehnike programiranja / Rezidentni programi
NA SOPSTVENI RIZIK



I ovog puta „grupni portret sa damom" na naslovnoj strani u znaku je naše nagradne igre. Predstavljamo vam softverski komplet „Budite Windows programeri" koji (u dva primerka) daruje firma „Atlantis" iz Ljubljane. Hardver bi mogao da bude zgoditak u sledećoj nagradnoj igri – računar 486 firme „Most" iz Beograda. Dama je *Tatjana Dragojević* (maneken iz Beograda), a sliku vam daruju *Nenad Marjanović* (fotografija) i *Darija Miloš* (make up).

52 Tehnike programiranja
POSTANI WINDOWS PROGRAMER

58 Tehnike programiranja / C++
KLASE - RAZGRANIČAVANJE GRANICA

61 Tehnike programiranja / PCL 5.0
OSNOVNE KOMANDE

64 Tehnike programiranja / Asembler
FORMIRANJE STABLA

88 Savetnik / Windows
WINDOWS I TASTATURA

69 Savetnik / Monitori
RAD SA DVA MONITORA

70 Dejan Ristanović
BAJTOVI LIČNE PRIRODE

73 Zoran Životić
MOJA ŠKOLA C-a

74 Dejanove pitalice
TEŽAK POREDAK

Na osnovu Mišljenja Sekretarijata za informacije SR Srbije broj 413-01-46 od 4. 2. 1991. na ovaj časopis plaća se osnovni porez na promet po povlašćenoj tarifi od 3 odsto

U PETOJ BRZINI

Na vrhu PC piramide do sada je neprikosnoveno stajao računar sa procesorom 80486 na taktu od 25 MHz. U računarstvu se stvari, međutim, menjaju veoma brzo – ovog proleća probijen je zvučni zid na oba, i na i386 i na i486, fronta, a prvi primerci ovih nadzvučnih mašina našli su se početkom maja i na našem ispitnom stolu.

Dejan V. Veselinović

GAMA AMD 386/40

Ovo je valjda prvi takav uzorak koji se pojavljuje testu. U osnovi, radi se o računaru koji je sasvim nalik na „standardnu" tri-osam-šesticu, sa istim delovima, ali sa dve bitne razlike. Prva je da umesto Intel procesora srce ove ploče čini jedan klonirani procesor američke firme AMD (Advanced Micro Devices), a druga je u radnom taktu, koji nije 33 nego 40 MHz.

Sa tim u vezi, jedna mala anegdota: ovaj računar je bio izložen na beogradskom sajmu tehnike na štandu firme Gama Electronics Beograd. LED prikaz na „tower" kutiji je sasvim ispravno pokazivao stvarni radni takt, ali su posetioci krajnje uporno odbijali da prihvate da računar stvarno radi na tom taktu, duboko ubeđeni da je prikaz, kao i mnogo puta do sada na našim sajmovima, bio podešen prema nekom SPEED ili LANDMARK test rezultatu. Da je tako, onda bi trebalo da pokazuje 65 MHz (ovo je bilo za ljubitelje tih testova).

Da pogledamo brzo o čemu se radi, bar što se standarda tiče. Tu je VGA video karta sa Tseng Labs ET4000 video čipom i 1 MB video memorije, što dozvoljava rad do rezolucije od 1024×768 sa 256 boja, sa ili bez preplitanja. Zatim je tu AT Bus (IDE) veznik i niz veznih linija (serijske i paralelne). Flopi disk od 1.2 MB i tvrdi disk Quantum AT105 od 105 MB, kao i jedna Cherry ASCII tastatura, kompletiraju spisak uobičajenih delova. Idemo na nestandardne stvari.

Matična ploča

Prvo i osnovno, tu je tajvanska matična ploča sa novim procesorom. Ovako veliki radni takt je uslovio izbor paketa vezne logike, jer koliko je nama poznato, jedino firma OPTI ima čipove predviđene za rad na ovom taktu, a najavljena su i verzije za 50 MHz. Procesor kao procesor, a jedino što na njemu otskače su slova AMD. Izrađen je u CMOS tehnologiji sa međurazdaljinama veza od 0.8 mikrona (!), što se ogleda u većoj pouzdanosti na velikim brzinama i manjoj temperaturi koju odaje u radu, što se opet prevodi na veću pouzdanost na duži rok. Po svemu ostalom, ova ploča je u celosti nalik na druge slične ploče koje smo toliko puta videli.

Radni takt ove mašine je 21% veći od do sada najbrže matične ploče u ovoj klasi, odnosno uopšte u klasi IBM kompatibilnih računara ako se izuzmu razni prototipovi i demonstracioni uzorci. Ono što je nas veoma zanimalo, obzirom da svakodnevno koristimo sličnu ploču na 33 MHz, jeste da li ćemo osetiti tu razliku (izmerićemo je sigurno). Odgovor na ovo pitanje ne zavisi samo od pukog takta procesora, već od čitavog niza stvari. Recimo, što više raste radni takt, sve je teže sinhronizovati ostatak računara sa sve bržim procesorom. Kvalitet vezne logike postaje sve značajniji sa rastom takta, a isto važi i za kvalitet BIOS-a.

BIOS je inače potpisala (bar nama) nepoznata firma Microid Research i nosi naziv Port OPTI324, a veoma je taze, jer oznaka navodi ovu godinu kao početnu.

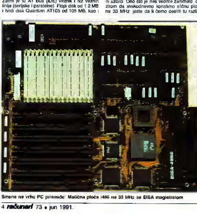
Performanse

Kako smo i očekivali, u rezultatima merenja ova ploča se pokazala veoma dobro. Prilikom ispitivanja, i prema potrebama samog merenja, probali smo čitav niz programa i moramo odmah reći da nismo naleteli ni na jednu nekompatibilnost ili bilo šta što bi ličilo na problem. Jednostavno, sve je radilo potpuno isto kao i na našoj R/Optima 386 Mk.II mašini, što će reći brzo i glatko.

Što se razlike u brzini rada tiče, moramo reći da smo na sopstveno čuđenje ipak osetili razliku. Kopajući malo dublje, pokazalo se da ta razlika ne potiče toliko mnogo od same brzine procesora, gde je nismo nešto posebno primećivali, koliko od sprege OPTI čipova i BIOS-a. Ta sprega, očigledno veoma efikasna, dovela je do video indeksa u tekstualnom režimu koji 95% veći, a u grafičkom režimu nešto skromnijih 66% veći. To verujte i te kako osetite, a pošto se u oba slučaja radi o istovetnim video kartama, razlika očigledno potiče od ploče. Indeks diska je u slučaju AMD ploče takođe bio nekih 16% veći, ali to je naravno pre svega zahvaljujući odličnom Quantum disku, koga i od ranije znamo kao veoma brzog.

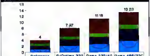
Video

Druga vanserijska (nadamo se, ne za dugo) stvar kod ovog sistema je Philips 7CM3279 monitor. Radi se o monitoru sa ekranom od 14 inča (36 cm) po dijagonali, sposobnim da radi do rezolucije od 800×600 bez i do 1024×768 sa preplitanjem. Standardno. Nestandardna je slika, koja po svom kvalitetu odstupa od proseka, pa i od konkurencije; slučajno smo imali prilike da je na istom stolu i sa istim kartama uporedimo sa jednim AOC CM-314 i jednim Targa (CTX) monitorom. Iako je bilo videti da je istina jeftiniji AOC monitor za dobru klasu ispod, a da je naš ljubimac Targa/CTX na velikoj muci da ostane ljubimac. Na kraju je u korist Philips-a prevagnulo to što ima marginalno manje ugaona izobličenja i osetno stabilniji prelaz iz jednog režima u drugi. Druga posebnost ovog monitora je što spada u najnoviju generaciju uređaja sa veoma malim stepenom zračenja na sve strane, a posebno unapred, odnosno ka korisniku. Postoji i drugi model sa oznakom 7CM3209 (razlika je samo u pretposlednjem broju), koji je nešto jeftiniji, ali zato nije zaštićen od zračenja. Inače, oba ova monitora se u svetu reklamiraju sa oznakom „Brilliance", koja bar ovom dobro stoji. Ime zaista



Smena na vrhu PC piramide: Matična ploča i486 na 33 MHz sa EISA magistralom

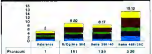
Slika 1



	Referensa	R/Optima 388	Gama 388/40	Gama 486/33C
Video	1	2 74	4 32	4 07
T disk	1	1 01	1 29	1 38
Memorija	1	2 03	2 2	3 88
Procesor	1	2 16	2 76	3 73

Mašinski testovi

Slika 2



	Referenca	R/Optima 388	Gama 386/40	Gama 486/33C
Proracuni	1	1 81	1 38	3 26
CAD	1	2 33	2 07	4 83
B podataka	1	1.18	1.23	2.1
P grafika	1	1 82	2.23	2.18
O tekste	1	1 37	1 28	2 77

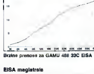
Programski testovi

Slika 3



Brzine prenosa za GAMU 386/40

Slika 4



Brzine prenosa za GAMU 486 33C EISA

odlične boje i veoma veoma oštru sliku Igla na metru za zračenje nam se nije ni pomerila

Hardverski indeks

Sve u svemu, hardverski indeks ove ploče je iznosio 2 79 u odnosu na referencu (NEAT ploča na 16 MHz 16-bitna VGA grafika) i bio je nekih 40% veći od R/Optime, pre svega zbog videa a pomalo i zbog diska (R/Optima opet vodi po ceni i kapacitetu diska jer ima 75% veći disk) U programskim testovima R/Optima je prošla mnogo bolje pre svega zbog svog ugrađenog koprocesora IIT 3C87 koji joj je obezbedio prednost u testovima sa CAD-om i proračunima Na žalost za sada su samo najavljeni koprocesori na 40 MHz, pa nismo bili u mogućnosti i da ih probamo

To ujedno eliminiše ovaj računar iz borbe za prvo mesto u „mlevenju brojki", ali ga čini apsolutnim favoritom bez konkurencije za recimo DTP gde je potrebna čista brzina procesora Istina je da će 486 raditi još brže ali je takođe istina i to da će koštati mnogo više a dobićete i ugrađeni koprocesor (bar do pojave verzije 486SX) koji vam ne treba Uopšte za sve delatnosti u kojima vam ne treba koprocesor obavezno razmotrite ovu ploču jer ona košta jedva oko DEM 100 više od standardne ploče na 33 MHz dakle plaćate oko 6 7% više za performanse koje su bar tri puta veće od doplate, a to je uvek dobar posao

GAMA 486/33C EISA

Ovo je prva mašina sa EISA magistralom a ujedno i najbrža mašina sa kojom smo radili uopšte, dva puta „naj" nije baš često, ali eto desi se

EISA magistrala

O EISA magistrali bi trebalo reći par reči tek podsećanja radi Ideja iza ove sabrnice je dosta složena sa jedne strane, ona mora da bude unazad kompatibilna (odnosno da može da primi i današnje ISA kartice), ali i da nudi nekoliko važnih razlika u poređenju sa standardnom ISA sabirnicom kao daleko veću brzinu (do 33 megabita u sekundi) veći širinu (32 umesto 16 bita) sposbnost potpune arbitraže (omogućavanje nekoj kartici u utičnici odnosno na magistrali da preuzme kontrolu nad računarom), itd U nekom najopštijem smislu EISA je namenski isto što i IBM-ov mikrokanal (MCA) ali za razliku od njega unazad je kompatibilna

Kontroler diska

Međutim nemajmo iluzija, osnovni motiv kod EISA sabirnice je brzina Zato nam je scre i skočilo kada smo u mašini videli EISA disk kontroler (inače konsti ESDI format) firme Ultrastor samo videti ovu karticu je zanimljivo jer je toliko puna elektronike u najsavremenijoj verziji površinskog lema (SMT) i mikro čipova a opet ih ima na obe strane kartice, koja je pune dimenzije AT formata Neverovatno mnogo elektronike odavno nismo videli toliko čipova na jednom mestu Po broju ima ih na kartici bar dva puta više nego na matičnoj ploči računajući i tri VLSI kola Na žalost ima i šest žica koje uspostavljaju veze tamo gde je štampa

Kontakt adresa

GAMA
Mišarska 11
YU 11070 Novi Beograd
Tel (011) 332-275 fax (011) 335-902

omanula ili naknadne veze kojih su se projektanti setili tek kada je štampa bila gotova

Pored brda čipova na kontroleru odskaču i pomalo neuobičajeni memorijski čipovi namenjeni keširanju u tzv ZIP pakovanju (tanko duguljasto pakovanje sa samo jednim centralnim redom nožica) Nose oznaku Toshiba 524256-10 (CMOS 4×256 Kb 100 nS) a u saradnji sa drugim memorijskim čipovima na pločici, daju (valjda) 1 MB keš memorije Pošto nismo imali dokumentacije ne znamo tačnu količinu keš memorije

Pored ovih čipova na pločici dominira i jedan Intel 82355 čip koji predstavlja vezu sa ostatkom paketa na matičnoj ploči i omogućava preuzimanje kontrole nad sabirnicom

Matična ploča

Sama ploča pored procesora 80486-33 ima postolje i za Weitek 4176 numerički koprocesor a veznu i pomoćnu logiku omogućava poseban paket čipova od samog Intel-a Intel je jedna od firmi koje su se učlanile u grupu proizvođača koja je osmislila i koja razvija EISA standard (pored Compaq-a Epson-a AST-a i drugih) Dimenzije matične ploče su kao kod standardne AT ploče pune veličine a odmah pokraj utičnica ima mesta za 16 SIM modula U njih možete staviti module od 256 KB 1 MB i 4 MB što znači da na samu ploču možete strpati do 64 MB RAM a U našem slučaju bila su popunjena samo četiri mesta modulima od 4 MB pa je sistem sadržao „samo" 16 MB

Pored osnovne sistemske i unutar čipa ugrađene keš memorije (skromnih 8 KB) na ploči smo našli i 128 KB drugostepene keš memorije Ovo je opet srednje vrednost jer drugostepena keš memorija može imati 64 128 ili 256 KB Što se DOS a tiče, pitanje je da li je ona uopšte i potrebna, ali u slučaju većeg broja korisnika (kada računar ima ulogu servera) ili drugih i razvijenijih upravljačkih sistema kao recimo UNIX a veća keš memorija će sigurno dati i bolje rezultate Ipak i ovakvi kakvi su odlični su

Ostatak sistema

Ostatak računara se sastojao od jednog Fujitsu disk od 700 MB dve disk jedinice (5 25 i 3,5 inča) jedna Morse KP1024 VGA kartice (Tseng ET4000 video čip, 1 MB do 1024×768 sa 256 boja sa ili bez preplitanja) i jedan I/O kartice Na žalost sem disk kontrolera sve ostale kartice su bile prema ISA standardu no ipak je bilo lepo konstiti ih makar da se uverimo da i ove kartice lepo rade u EISA utičnicama

ZVEZDA PONUDE

Procesor i486 više nije neka novost – gotovo da i nama prodavca kod nas i u svetu koji je nudi bar jedan model računara, ne računajući permutacije sa diskovima i video sistemom, koji se oslanja na upravo ovaj procesor. Tim koracima je krenulo i beogradsko preduzeće „Most", a mi smo malo „pretresli" njihov krunski model.

Dejan V. Veselinović

Osnovna koncepcije ovih i ovakvih računara jeste da konsniku ili konsnicima obezbedi relativno veliku računarsku snagu za relativno skromnu cenu. Iako su ove mašine sve samo ne jeftine, kada se uzme u obzir njihova moć, videće se da zapravo i nisu tako skupa. Naravno, tu je veliko pitanje i šta dobijate za vaše pare, ne samo u smislu delova i performansi, već i ostalih aspekata kvaliteta koji nisu tako lako uočljivi. Ovo su neke stvari na koje se kod nas do sada nije mnogo obraćala paznja, ali sa polaganim sazrevanjem tržišta i kupci normalno postaju selektivniji. Drugim rečima, cene sve manje predstavlja sve i svja, a sve više pažnje se poklenja odnosu kvalitet/cena.

Da bi postigao povoljan odnos kvaliteta i cene, svaki prodavac mora da reši jednu dilemu: sa jedne strane, trebaju mu kvalitetni a samim tim i skupi delovi, a sa druge strane, zbog cene, trebaju mu veoma povoljni izvori tih delova. Vezni materijal između njega i kupaca je kvalitet izrade i ostale prateće usluge, kao što su garancija, brzina odziva i tome slično. Imajući sve ovo u vidu, bacili smo se na testiranje računara sa oznakom „Most Star", što valjda znači zvezda ponude.

Matična ploča

Bilo nam je veoma drago da u računaru naiđemo na matičnu ploču firme „Protech", čiju smo matičnu ploču pre nekih 4 godine i sami imeli. Ima format „baby AT" ploča, odnosno manjih dimenzija. Elementi na ploči su očigledno rezultat novijeg razvoja ove tehnologije: pored samog procesora postoji i prazno podnožje za Weitek 4167 numerički koprocesor. Vezna logika je delegirana na OPTI paket od dve čipa, podržana trećim od Texas Instruments-a.

OPTI (paket sa oznakom HiB/486) čipovi u poslednje vreme postaju dosta česta pojava na tajvanskim matičnim pločama. Paket se sastoji od dva VLSI čipa, sa strogo podeljenim delatnostima. Prvi čip, sa oznakom 82C481, je kontroler AT magistrale, odnosno veznik između proceora i magistrale. Sadrži kontrolnu logiku procesora, kontrolnu logiku magistrale, sinhroni delitelj radne učestanosti i kontroliše lokalnu magistralu sa periferalima. Naravno, u sebi sadrži i potrebnu logiku za konverziju 8-, 16- i 32-bitnih kartica, jer se sve tri vrste mogu naći istovremeno na magistrali, a efikasnosti radi, u praksi će postojati bar jedna 32- i 2 16-bitne kartice (keš memorije, video i I/O kartica).

Drugi čip, sa oznakom 82C482 („C" označava izvedbu u CMOS tehnologiji) takođe ima čitav niz funkcija. Prvo i osnovno, on omogućava kompatibilnost sa normalnim (tj. 16-bitnim) AT okruženjem, kako bi se obezbedila kompatibilnost sa IBM AT okruženjem, ovo je osnov svake kompatibilnosti. Drugo, ovaj čip se bavi svim radom sa memorijom i vezom keš-sistemska memorije. Ovo poslednje ima čak dva aspekta: prvi se odnosi na preplitanje sistemske memorije, kako bi se ubrzao rad sa njom nezavisno od keš memorije (pa kada i dođe do „promašaja", da očitavanje sistemske memorije bude što manje usko grlo), i drugo, radom u tzv. režimu udara (burst mode), kada dođe do protoka informacija koji je veći od onoga što sistem podržava u trajnom stanju.



Snimio: Nenad Marjanović

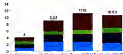
Teška artiljerija za stono izdavaštvo: Most i486

Keš memorija

Usputni efekat ovakve organizacije (keš memorija-ispreplitane sistemske memorija) pored povećaja brzine, je i smanjenje cene ukupne memorije. Keš memorije koristi veoma brzu (25 nS i manje) statičku memoriju, koja je osetno skuplja od obične, sa druge strane sporija sistemska memorija je ujedno i jeftinija, pa delimično kompenzuje cenu statičke memorije. U ovom konkretnom slučaju, to postaje važnije od proseka jer i količina keš memorije veća od standarda (ako ga uopšte ima), koji najčešće iznosi 64 KB. U slučaju računara „Most Star", drugostepana odnosno spoljna keš memorija (podsećamo da procesor 80486 ima ugrađenih 8 KB keš memorije) je dvostruko veća i iznosi 128 KB, proširivo do 256 KB.

U nekoliko sličnih prilika, pomenuli smo da je sama količina keš memorije daleko manje značajna od kvaliteta algoritma za keširanje organizacije keš memorije i, najzad, brzine sistemske memorije. Jasno je da su se projektanti OPTI čipova veoma pozabavili svim ovim aspektima, pa u slučaju postojanja parnih ban-

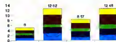
Slika 1



	Referenca	R/Optima 486	Gama 386/40	Most 486/25
Video	1	3.14	4.92	3.88
T disk	1	1.01	1.29	1.53
Memorija	1	2.06	2.2	3
Procesor	1	2.81	2.78	2.42

Mašinski testovi

Slika 2



	Referenca	R/Optima 486	Gama 386/40	Most 486/25
Proracuni	1	2.42	1.38	2.38
CAD	1	3.78	2.07	3.88
B podataka	1	1.18	1.23	1.76
P grafika	1	2.29	2.23	2.4
O teksta	1	2.88	1.26	2.1

Programski testovi

ki memorije (2×4 SIM modula, pošto se radi o 32-bitnom a ne 16-bitnom pristupu memoriji, broj modula u jednoj banki raste sa 2 na 4), sistem radi sa preplitanjem, a u slučaju neparnog broja samo sa straničnim pristupom. Pored povećanja brzine, ovo ujedno produžava vreme pristupu memorije za najmanje 15 nS, a u nekim slučajevima i za više. Dakle, može se staviti i nešto sporija memorija, čime se ujedno i štedi po jednom modulu, to možda i nije neka ušteda, ali na 8 ili 16 modula razlika postane sasvim lepa suma.

U našem slučaju, u računaru su dve banke SIM modula bile popunjene (čime se dobilo preplitanje), a veličina keš memorije je bila 128 KB. To su dobre vesti, loša vest je obezbedila logika projektanta, koji je na ploču smanjenih dimenzija ipak stavio 8 utičnica za proširenja, ali odmah nakon toga korisniku oduzeo jednu, a možda i dve time što se keš memorije nalazi na posebnoj kartici u posebnoj utičnici, a druga takva posebna utičnica služi daljem širenju memorije preko 8 MB. Istina, u međuvremenu, ova druga utičnica se može koristiti kao standardna 8-bitna utičnica, ali na ovakvim mašinama, kod kojih uopšte nije nebulozno očekivati i više od 8 MB memorije, to je ipak bar relativna mane.

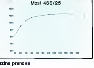
Što se rada tiče, ploča je čista kao suza, bez ikakvih žica i krpeža u poslednjem momentu. Sve je lepo poređeno i uredno zalemljeno, a krajnja obrada je odlična.

Ostale elektronika

Od ostalih periferija, tu je sve što treba da bude tu. Prvo, flopi disk jedinice u oba formata, 5,25 i 3,5 inča, su poreklom od japanskog proizvođača firme TEAC (FD-55GFR i FD 235HF), obe su poznate i sa dugim stažom u proizvodnji, što ih čini pouzdanim i dobro poznatim, čak i u slučaju kvara. Tvrdi disk je jedan od modela firme Conner, sa oznakom CP 3204F i kapacitetom od 180 MB. Pored zaista odlične brzine mehanike, ovaj disk ima zaista najbolju brzinu prenosa koju smo ikada videli na IDE ili AT Bus disku. Ovo se pre svega odnosi na ravnomernost prenosa, a dodamo li tome i veoma velike brojke, onda znamo da se radi o zaista vrhunskom modelu. Pun pogodak u smislu izbora.

Najzad, video karta je poreklom od nama potpuno nepoznatog proizvođača, američke firme Diamond Computer Systems, a model se zove „Speedstar". Po oznaci na pločici, već je u četvrtom izdanju, a za osnov koristi poznati

Slika 3



Brzina prenosa

Tseng ET4000AX čip, o kome smo se već nekoliko puta veoma pozitivno izjasnili (i upotrebili ga u svim projektima R/Optima). Jedino što se nama lično ne dopada na ovoj kartici je način na koji se dobijaju razna učestanosti za video režime. Umesto pojedinačnih kristala, ova kartica koristi samo jedan oscilator na 40 MHz i čip sa oznakom ICS 1394, koji od te jedne učestanosti izvlači ostale. Nama se ovo ne sviđa, ali mi smo verovatno već pomalo staromodni, karta je inače radila bez ikakve greške, pa i bez ikakvih vidnih deformacija.

Monitor

Ovaj put, prava zvezda programa je zapravo monitor. Ovo pre svega zbog njegovih, blago rečeno, impozantnih dimenzija: po dijagonali, ekran ima ravno 20 inča, odnosno 51 cm. Radi se o modelu firme Hitachi sa oznakom CM2085MU. Ovakvi monitori zaista nisu za narod, maksimalna rezolucija ide do 1280×1280 i to bez ikakvog preplitanja. Naravno, da bi to bilo moguće, propusni opseg mora biti veoma širok, a sa opsegom do 80 kHz (BNC veznici, horizontalno skaniranje do 64 kHz) po horizontali on to i jeste. Ako slika treba da bude stabilna, onda i vertikalno osvežavanje mora da bude veliko, podsećamo da se proizvođači monitora za smrtnika veoma hvale brojkama od 70 Hz naviše, a negde na kraju uputstva, u tehničkim specifikacijama, Hitachi spominje brojku od 50-120 Hz.

Kontakt adresa

MOST
11000 Beograd, Kralja Bodina 2
Tel (011) 422-170, Fax (011) 426-144

Rezultat svega toga je slika koja je sem velika, sem zaista kristalno jasna, uprkos relativno velikim tačkama od 0.31 mm. Ovo poslednje je pomalo čudno, čovek bi ipak očekivao tačke ne veće od 0.28 mm, a verovatnije 0,25 mm, od ovako velike tačke očekivala bi se pomalo gruba slika. Po nama, ona nije gruba, mada smatramo da joj nedostaje izvesna finesa, no, ovo je i te kako subjektivna stvar, drugima će slika delovati neverovatno. Istina, ona pleni veličinom. Oni kojima ovakav monitor treba za CAD će se sigurno razboleti kada vide monitor, što zbog njega, što zbog cene (81 000 dinara), u svakom slučaju treba ga pogledati. Cela redakcija je gledala sliku repa aviona u rezoluciji od 1024×768 u 256 boja, stvarno je sjajno izgledalo, pa smo jednog člana posle na silu odvlačili od monitora.

Performanse

Kao što se moglo pretpostaviti, performanse ove mašine su na zavidno visokom nivou. Sve radi veoma, veoma brzo, a posebno ističemo tvrdi disk, koji je postigao daleko najbolje rezultate od svih IDE diskova do sada, prevazišavši čak i Quantum diskove.

Što se video performansi tiče, one su ipak malo podbacile, to nas je začudilo, jer smo ipak nekako očekivali više. Da se razumemo, performanse nisu loše, ali obzirom na veliku keš memoriju, nekako smo očekivali više. Sve u svemu, nema većih iznenađenja.

Sve u svemu, radi se o jednoj dobroj mašini, sa dobro odabranim delovima. Monitor je dosta skup, ali ako vam ne treba baš ovako veliki ekran, odlukom za neki manji ali i dalje dobar monitor možete uštedeti dobrih šezdesetak hiljada dinara.

Iako je rešenje sa keš memorijom u posebnoj utičnici po nama rasipničko što se utičnica tiče, ploča je u celini gledano dobra. Tvrdi disk je izvanredan, a video karta sasvim pristojna, pa iako ne briljira, ne može se reći ni da ruši koncepciju.

Sve u svemu, za cenu od oko 200 000 dinara, dobijate jednu dosta jaku mašinu koja, iako dobra, ipak ne uspeva da zapali strasti. U celini gledano, mi bismo se pre odlučili da cenu oborimo za kojih 55 000 dinara, zamenivši ovaj (pre)veliki monitor za neki jednako dobar ali manji ("Most" nudi i Sony monitore, i to za sasvim razumnu cenu). Disk bismo obavezno zadržali jer je sjajan, ploča je dobra, a sa cenom od oko 140-150 000 dinara (sa manjim monitorom), i cena je dobra za ovu klasu.

SAVRŠEN PAR

Prenosivi računari formata A4, čak i sa procesorom SX, više nisu neka retkost, bez obzira što kod nas dolaze sa već poslovičnim zakašnjenjem. *Notebook* štampači, međutim, predstavljaju prvorazrednu novost. Najzad, obe stvari, koje zajedno čine prenosivu kancelariju, su do sada neviđene bilo gde. Probali smo jednu takvu kombinaciju za vas.

Dejan V. Veselinović

Oba uređaja su zaista mala, taman toliko da stanu u jednu normalnu aktn tašnu, istina bez nekih dodataka, kao što je recimo punjač baterije za računar ili štampač, jer i on radi na baterije. Zajedno ova dva uređaja su teška celih 7,0 kilograma, ili nešto malo više od našeg referentnog računara Chicony modela LT 3600. Veličina paketa je 440×290×60 mm. Ali, to nije sve: štampač, verovali ili ne, emulira Hewlett-Packard Laserjet II! Sa celom rezolucijom od 300×300 tačaka po inču. Podimo redom.

LIGHTLINE NOTEBOOK

Skromne dimenzije (280×220×53 mm, ŠxDxV) i skromna težina od svega 3,2 kg stavljaju ovaj računar u klasu definitivno prenosivih mašina. Po broju opcija koje nudi i po standardnim osobinama može se reći da spada u gornji deo ponude ovakvih mašina u svetu danas, iako ni po čemu nije izuzetan. Radi se o mašini na bazi procesora Intel iAPX 80386SX na radnom taktu od 16 MHz, standardno je opremljen sa 2 MB memorije u vidu dva SIM paketa sa brzinom memorije od skromnih 100 nS. Kada smo već kod toga, memoriju možete proširiti za još jednom toliko, što nije neobično, ali lakoća sa kojom to možete sami uraditi jeste neobična. Treba da odlepite nalepnicu na prednjoj gornjoj strani mašine, odšrafite jedan jedini šraf i naprosto ubacite dva SIM modula. Tu je i poseban odeljak za modem, koji zahteva da bude od istog proizvođača. Ovo je do sada najjednostavniji metod proširenja, koji smo videli na bilo kom prenosniku.

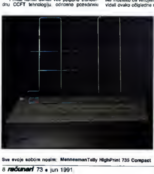
Prikaz koristi danas već potpuno standardnu CCFT tehnologiju, odnosno pozadinski osvetljen super uvrnuti tečni kristal. U odnosu na referentni model ovaj prikaz, takođe sa VGA rezolucijom, ima nekih prednosti, ali i jednu krupnu manu. Od prednosti tu je pre svega činjenica da na njemu ima daleko manje izraženog efekta curenja („bleeding"), odnosno vertikalnih i horizontalnih linija koje se javljaju kao posledica preklapanja segmenata sa kristalima. Pošto je ekran fizički manji, a broj segmenata isti, dobijamo sličan efekat kao kada poredimo dva televizora, jedan sa većim a drugi sa manjim ekranom, pa nam izgleda da onaj sa manjim ekranom ima oštriju sliku. Odnos horizontale prema vertikali (tzv. aspect ratio) je idealan i iznosi 1:1,34. Nominalna površina ekrana je oko 227 cm², dok je kod referentne mašine površina oko 340,6 cm², ili za 50% veća.

Ono što kod ovog ekrana ne valja je činjenica da uprkos razumnim dimenzijama (imajući u vidu ukupne dimenzije mašine) na gornjoj i donjoj ivici ekrana postoji po dobar prst praznog prostora koji se ne da iskoristiti. Dalje, stepen najvećeg mogućeg osvetljenja ekrana (ne kontrast) je takav da ekran postaje teško čitljiv u iole svetlijem ambijentu. Recimo, na taj problem smo naišli u jednoj svetlo (i sveže, još nas bole leđa) okrečenoj prostoriji pod isključivo dnevnim svetlom i to samo sa reflektovanom svetlošću. Ovo su pomalo neshvatljive mane za današnje vreme, ali i sigurna indikacija da je računar prilično brzo pripremen za tržište. Ne možemo da verujemo da i konstruktori nisu videli ovako očigledne nedostatke.

Treći problem u vezi ekrana se pojavio na jednom jedinom mestu, a to je u programu Harvard Graphics. Pošto se radi o najnovijoj verziji (2.3.01) i pošto nikada ranije (pa ni sa starim verzijama) nismo naišli na ovaj problem, navodimo ga informacije radi. Računar je jednostavno odbio da napravi sliku u VGA rezoluciji, koju je uporno prikazivao sa odsečenim donjim delom slike i u rekli bismo CGA rezoluciji. Kada smo sve ponovili sa EGA rezolucijom, bilo je u redu, ali EGA je EGA, a ovaj se prodaje sa VGA grafikom. Pošto smo sve morali da obavimo sa EGA rezolucijom, rezultati merenja su nužno pomereni u odnosu na standarde. Istini za volju, svugde drugde je prikaz sasvim normalno radio kao VGA, mada sa pomenutim praznim prostorima na vrhu i dnu ekrana. Ovo dakle navodimo kao elemenat upozorenja, jer ako se desi jednom i to u programu koji nije ni malo marginalan, postoji mogućnost da se desi opet negde drugde.

Ovaj računar je bogat raznim vezama. Sa leve strane se nalazi veznik za spoljni monitor, sa desne veznik za spoljnu tastaturu, a pozadi paralelni, dva serijska i veznik za proširenje. Svi veznici, sem onog za tastaturu, imaju poklopce, čiji smo nedostatak kritikovali kod drugih prenosnika, a u ovom slučaju hvalimo.

Tastatura je dosta pristojna u poređenju sa Commodore prenosnikom, iako je i ova membranska, a sa referencom se i dalje niko ne može meriti po kvalitetu tastature. Ono što je bar nas posebno iritiralo kod ove tastature je katastrofalan nemački raspored tastera. Kada se broj tastera smanji zbog dimenzija računara, pa tasteri moraju imati i po tri funkcije, a onda neko na svu tu muku još uglavi ionako

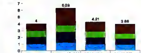




Snimio Nenad Manjanović

Sve svoje sobom nosim: MannesmanTally HighPrint 735 Compact | Lightline notebook

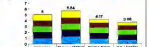
Slika 1



	Referenca	Chicony LT3600	Nibble Nylap	Ibis Lightline
Video	1	2.61	1.36	1.37
T disk	1	1.06	0.88	0.8
Memorija	1	1.5	0.83	0.83
Procesor	1	1.23	0.85	0.86

Mašinski testovi

Slika 2



	Referenca	Chicony LT3600	Nybble Nylap	Ibis Lightline
Proracuni	1	1.16	0.83	0.85
CAD	1	1.02	0.77	0.73
B podataka	1	1.05	0.57	0.65
P grafike	1	1.13	0.8	0.65
O teksta	1	1.28	0.6	0.66

Programski testovi

šašavu nemačku tastaturu, a onda imate katastrofu da se snađete. Posle tri dana rada mi još uvek nismo našli neke znake. Ako tastatura treba da predstavlja zagonetku, onda je proizvođač u celosti uspeo. Prijavićemo je SUP-u, da vidimo, možda će oni imati više sreće od nas.

Dakle, ako uzimate ovakav računar, nemojte da vas obrlate za bilo šta sem standardne ASCII tastature.

Što se performansi tiče, u celini gledano one su sasvim pristojne. Odmah da napomenemo da bi sa dopunska dva SIM modula one bile 15-20% bolje, jer bi računar mogao da preplиće memoriju sa dve banke od po dve modula, dok je ovako radio sa punim stanjem čekanja. Čak i tako, performanse su sasvim pristojne, posebno u kontekstu klase ovog računara.

O vrsti kupaca za ovaj i ovakve proizvode nema smisla mnogo govoriti, jer se zna ko bi to mogao biti, od poslovnog sveta, preko prodavaca koji bi modemom mogli da se povežu sa računarima u svojim firmama, do pojedinaca koji žele mali, kompaktan, pa čak i jedini računar uz uslov prenosivosti. Istina, tvrdi disk od 20 MB nije baš veliki, ali verujemo da će postojati i verzija sa duplo većim diskom. Kod kuće možete držati spoljnu tastaturu i monitor.

U celini gledano, ovaj proizvod ne bismo preporučili, bar ne sa nemačkom tastaturom. Sa standardnom ASCII tastaturom bi već nekako moglo, ali ponavljamo da je prikaz ispod standarda današnjice. Brine nas i nesrećan slučaj sa *Harvard Graphics*-om, kao mogući indikator daljih sporadičnih problema. Ove zamerke po nama zasenjuju vrline lakoće instalacije modema, proširenje memorije i mogućnost ubacivanja numeričkog koprocesora (to mora da uradi prodavac, jer računar treba rastaviti).

MT HIGHPRINT 735 COMPACT

Svakih godinu dana ili dve pojavi se neki proizvod koga zaista treba videti da biste verovali da je i to moguće. E, ovaj štampač je idealan primer upravo takvog proizvoda. Da ga nismo videli, ne bi verovali da i tako nešto može.

Dakle, dimenzije štampača su 290×244×59.5 mm (ŠxDxV), i to sa otvorenim držačem za papir, težina je 3,8 kg sa akumulatorom. Sva to vam mirne duše stane u jednu ruku, mada nepunih 4 kg i nije tako malo na duži rok.

Sa prednje strane se nalazi pet komandnih tastera i pet signalnih dioda. Tasteri su dosta standardni: prvi je za uključivanje i isključivanje, drugi je ON LINE/OFF LINE, treći je za



Nije baš kao laser...

FEED FORWARD, četvrti za COPY (na pritisak ponovo štampa poslednju odštampanu stranicu koliko pritisaka, toliko kopija), a peti ima tri pozicija i služi određivanju gustine štampanja u zavisnosti od kvaliteta papira. Signalne diode su za mrežu, ON LINE, proveru akumulatora, indikator punjenja akumulatora i upozorenje za papir i/ili traku.

Pošto smo štampač dobili nepripremljen za rad, imali smo priliku da vidimo kako se oseća korisnik koji ga po prvi put pokreće. Priručnik sa uputstvima je zaista veoma dobar i detaljan, čak i pomalo nemački isuviše detaljan. No, ako ga pažljivo pročitete na engleskom ili nemačkom, nećete imati problema. Najgore je ubacivanje trake, jer je to pipav posao, ali uz malo truda, po prvi put u životu, obavili smo to za desetak minuta. Štampač valja otvoriti, ali vam za sve to, a ni za bilo šta drugo što biste kao korisnik trebali da uradite, ne treba nikakav alat, čak ni šrafciger. Nemačka tehnika *par excellence*.

Delova iz Japana naravno ima na pretek, recimo, glava je proizvod firme Toshiba. Bez obzira na sva to, model 735 naprosto pleni svojom jednostavnošću i funkcionalnošću.

Koliko je funkcionalan, prosudite sami: od emulacija, navode se (očekivani) modeli Epson LQ-850 i IBM ProPrinter, ali i (neočekivani) model Hewlett-Packard DeskJet Plus i totalno neočekivani Hewlett-Packard LaserJet II. Baš tako, ovo malo čudo veoma uspešno emulira i laserski štampač sa svih 300×300 tačaka po inču. Reda radi, probali smo sve emulacije, ali je jasno da smo najveći deo vremena posvetili emulaciji lasera.

Pošto se u štampač može poslati nekoliko pisama koja mogu zauzimati i celih 178 KB (malo za laser, ogromno za ovakvog klinca),

Kontakt adresa

Ibis Sys
Pariskih komuna 22
YU-11070 Novi Beograd
Tel. (011) 609-650, fax (011) 699-081

probali smo i to i naravno, radilo je bez greška. Pored toga, pisma koja se standardno isporučuju u ROM-u štampača (Courier 12, 10 i 8.5 punkta, matematičko pismo od 10 i 8.5 punkta, Helvetica i Times Roman od po 12 punkta) su i više nego dosta. Eh, da lasere isporučuju sa takvim pismima.

Tajna je naravno u termalnoj glavi, koja ima 2 560 grejnih elemenata i traci, koja po deklaraciji izdržava oko 150 stranica. To što je kvalitet štampe na našem uzorku bio zaista loš nakon dužih provera i čišćenja, svelo se na traku kojoj je verovatno istekao rok trajanja. Na kutiji je jasno napisano na četiri jezika da se trake moraju upotrebiti pre isteka roka trajanja. Do tog zaključka smo došli nakon pregleda trake koja je čak počela da se smežurava. Šteta, velika šteta što nije bila „taze", jer onda bi rezultati bili impresivni.

Kao što rekosmo, ovo malo čudo morate da vidite da biste poverovali da i tako nešto može. Naravno, i ovaj štampač ima svoja ograničenja. Prva slaba tačka je sama traka, koja niti je posebno trajna, niti posebno pogodna za nameštanje, sa druge strane, ne vidimo kako bi ovako nešto bilo moguće bez upravo ovakvog rešenja trake i termalne glave. Kapacitet istina nije bog zna šta, ali se opet vraćamo na dimenzije, pa i namenu štampača. Nigde ne piše da on treba da zameni vaš laser (a u toj je klasi), ali probajte da ponesete laser na put, makar samo i do aerodroma.

Zbog toga što smo raspolagali sa samo jednom trakom, nismo se usudili da proveravamo trajanje akumulatora. Ako izdrži 50 stranica, to je i više nego dovoljno za većinu namena, utoliko pre što je utičnica sa ispravljačem mala i lagana, pa nije neki problem nositi je sa sobom. Traka je opet poveća, približno sličnih dimenzija kao i ona za štampače Epson FX, pa iako nije teška, nije ni mala. To je valjda i jedina koncesija prenosivosti koja je učinjena.

Najzad, pogledajmo kome je namenjen ovaj proizvod. Poslovnim ljudima svih vrsta, kratko i jasno. Od onih koji će na njemu odštampati izjavu za javnost o otkupu neke megakorporacije, do prodavaca recimo računarske opreme, koji bi u aktn tašni mogli imati ovaj štampač i mali računar, pa bi na licu mesta mogli izdavati fakture. Ako se tom opremom ne bi impresionirali potencijalne kupce, ne znamo šta bi još morali da ponesu.

Odličan proizvod, koji je preko noći zasenio sve što bi mu moglo i prići, bar do sada. Ovo je proizvod koji je sam za sebe posebna klasa, svaka mu čast.

FORMULA MEĐU DISKOVIMA

PSI (Perceptive Solutions) je tvrtka koja je stvorena 1987. godine. Reći ćete: „Mlada tvrtka!" No, ljudi koji u njoj rade su vrsni poznavaoci svog posla, te su u stanju zaključiti što tržištu treba. Put na tržište su prokrčili svojim keš kontrolerima tvrdih diskova. Naravno, na tržištu nisu sami, ali im to, izgleda, nimalo ne smeta.

Marko Raker

Na moju molbu da mi firma CC&T, zastupnik tvrtke PSI za Jugoslaviju, posudi za potrebe testiranje dva hyperSTORE keš kontrolera odgovoreno je pozitivno. Za koji dan na radnom stolu u mojoj sobi se našao poveći paket koji je sadržavao dvije glomazne kartice. Jedna, vrlo teška (zasigurno najteža koju sam dosada vidio), bila je hyperSTORE 1600 sa ugrađene 4MB RAM-e i ESDI mediadapterom, dok je druga, napravljena po najmodernijom *surface mount* tehnologijom bila hyperSTORE 400 IDE caching kontroler sa „svega" dva megabajta RAM-a. Obje kartice su pune dužine i maksimalnog formata, jedina uočljiva razlika je ta što je ESDI mediadapter montiran na hyperSTORE 1600 kao „sendvič" karta i to na samom njenom kraju suprotno od bus konektora kako bi oslobodila prostor za ostale kartice. Uz kontroler je stigla i disketa s pomoćnim programima, više nego opsežno i detaljno uputstvo za upotrebu, registracijska kartica te garancija. Također su stigli razni drajveri čak i za neke vrlo egzotične operativne sisteme, te pregršt propagandnog i materijala za štampu.

OSNOVNE KARAKTERISTIKE

HyperSTORE 1600 je, baš kao i model 400, stvoren oko procesora Zilog Z280 (sjetimo se Spectrumovog Z80), koji podržava i interni multitasking, koji je itekako iskorišten. Taj procesor, osim osnovne funkcije keširanja diska, radi i mnoge druge, možda čak i zanimljivije operacije. U svojoj maksimalnoj konfiguraciji hyperSTORE 1600 podržava nevjerojatnih 50.4 gigabajta magnetskog medija. Princip mediadaptera („sendvič kartice") omogućava da maksimalno proširujete svoje hard disk kapacitete, što znači da u jednom trenutku na istom kontroleru možete imati u bilo kojoj kombinaciji zastupljene MFM, RLL, ESDI, SCSI ili IDE diskove. Na kontroler možete paralelno spojiti 4 mediadaptera, što znači da maksimalno možete imati 8 MFM, RLL ili ESDI diskova, 16 IDE diskova (IDE mediadapter se proizvodi u verziji za 2 ili 4 hard diska), te 28 SCSI hard diskova. HyperSTORE 1600 podržava od maksimalno 20 MB RAM, tj. keš memorije koja se na kontroler ugrađuje pomoću SIMM modula, 4MB stanu na sam kontroler, dok sa ostalih 16 montira na posebnu karticu koja se sa kontrolerom povezuje putem dodatnog kabla.

Minimalna memorija koju zahtjeva kontroler je 512KB. Transfer podataka između kontrolera i računala je dotjeran do samog maksimuma ISA standarda, tj. 4MB u jednoj sekundi u kontrolerovom „native" modu ili 2.5 (a i više) u Western Digital kompatibilnom modu. Jedan od svakako egzotičnijih detalja na ovom kontroleru je njegov vlastiti „real time" operativni sistem. Za razliku od svih ostalih inteligentnih kontrolera, hyperSTORE nema operativni sistem pohranjen samo u ROMu, već i na posebno rezerviranom mjestu na hard disku. HyperSTORE također podržava i *mirroring* (automatski *backup* podataka na dvojnom sistemu diskova). ECC je sa standardnih 24 ili 32 bita povećan na 52 kako bi što bolje osigurao vaše podatke.

Naravno, hyperSTORE 1600, kao i hyperSTORE 400, ima na sebi ugrađen flopi disk kontroler.

HyperSTORE model 400 se razlikuje ne samo u tome što nije toliko modularan, već se isporučuje u verzijama ESDI, IDE i SCSI, te je maksimalna količina memorije samo 4MB. Rad kontrolera je apsolutno identičan, te možemo reći da je hyperSTORE 400 samo „okljaštrena" verzija kontrolera 1600.

OD KEŠA DO KEŠ KONTROLERA

Kao jedan općeniti pokazatelj možemo reći da se procesorska snaga u personalnim računalima od osamdesetih pa sve do danas udvostručava otprilike svake tri godine. Premda je razvoj paralelno pravljen i na ostalim dijelovima PC-a (na primjer video, memorija, hard diskovi), taj napredak ni izbliza ne prati razvoj mikroprocesora.

Prvi dio PC-a koji je posustao u buktajućem razvoju procesora bila je memorija. Da bi nekako sustigli tu razdaljinu između brzine memorije i mikroprocesora, tvorci PC računala su sa mainframe računala posudili ideju keš memorije. Keš je mala količina vrlo brze memorije kojoj je mikroprocesor u stanju pristupati sa svojom maksimalnom mogućom brzinom. Sadržaj keš memorije se konstantno obnavlja prema potrebama procesora. Proces obnavljnja sadržaja keš memorije je upravljan sa velikim brojem kompliciranih, potvrđenih matematičkih algoritama. Koristeći tu metodu keš memorija može poslužiti većinu (više od 90%) potreba mikroprocesora, smanjujući utjecaj spore DRAM memorije na rad računala.

Odmah nakon memorije pojavio se i problem masovne memorije (hard diskova). Tvorci mainframe i mikro računala su taj problem riješili na njima zadovoljavajući način pomoću takozvanog disk koprocesora. Taj koprocesor je obavljao isključivo poslove optimizacije i upravljanja hard diskova. Brzi pristup svakom podatku se rješavao pomoću takozvane „data striping" metode (kidanja podataka). Princip je bio taj da se svaki podatak cijepao na veliki broj malih podataka koji su se tada simultano zapisivali na obično 4 ili 6 hard diskova. Na taj način, kada bi korisnik zatražio određeni podatak, svih 4 ili 6 diskova su u otprilike istom trenutku krenuli čitati podatke da bi ih odmah zatim velikom brzinom poslali u memoriju.

Na PC-u je ta metoda bila neprihvatljiva, budući da je klasična PC konfiguracija neusporedivo manja od bilo kojeg mainframe-a, koji standardno ima po nekoliko gigabajta masivne memorije, koja se u tom slučaju redovno nalazi na nekoliko fizički različitih hard diskova. Takovi veći sistemi su često koristili nekoliko brzih I/O (ulazno/izlaznih) kanala isključivo da osiguraju brz prijenos podataka do diskova. Da bi mogli kontrolirati više ulaza i izlaza, rješenje je pronađeno u paralelnom rješavanju zadataka.

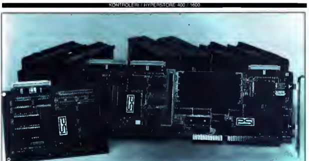
Da bi tvorci PC-a ostvarili svoj cilj koji se krio u jeftinom računalu, stvorili su vrlo jednostavan hard disk kontroler. Bit kontrolera je bila u tome da je njime (a i diskom) upravljao sam PC na štetu svojeg procesorskog vremena. Ovo je zaista vrlo jeftino rješenje, ali traži dodatnu procesorsku snagu. Za PC-e, XT-e i prve AT-e, zahtjevi aplikativnog softvera na hard disk su bili minimalni i nisu stvarali usko grlo. No, kod brzih 16, 20MHz i 33MHz računala brzine tvrdih diskova je počela da predstavlja ozbiljnu kočnicu. Na žalost, jedino rješenje koje se moglo preporučiti nezadovoljnim kupcima bilo je „Kupite brži hard disk". Kako na svetu nema dovoljno brzog tvrdog diska koji bi mogao da prati moderne procesore, inženjeri su posegli za alternativnim rješenjima. Jedno među njima se zove „keš kontroler tvrdog diska".

IZ USKOG...

HyperSTORE keš kontroleri su stvoreni da bi riješili usko grlo između hard diskova i memorije. Budući da rješenje sa mainframe i mikroračunala nije bilo zadovoljavajuće – krenulo se drugim putem. Prvi korak u rješavanju zadatka bilo je maksimalno smanjivanje i ubrzavanje sistemskog rada na kontroli hard diskova. Da bi na zadovoljavajući način riješili problem, inženjeri u PSI-u su iskoristili Z280, 16 bitni procesor sposoban za multitasking, koji može paralelno rješavati mnoge zadane probleme. Između ostalog, korištenje dodatnog procesora omogućava hyperSTORE-u da koristi i keš algoritme koji su samom računalu apsolutno transparentni, a ujedno i zaštićeni od mogućih konflikata sa raznim drajverima, TSR programima i slično.

Podaci su u RAM memoriju kontrolera spremljeni uz pomoć takozvanih „heurističkih" algoritama razvijenih u tvrtci PSI. Ti podaci se na zahtjev računala maksimalnom brzinom šalju iz memorije direktno u računalo štedeći pritom vrijeme koje bi inače hard disku trebalo da pronađe, pristupi, pročita i pošalje podatak. Naravno, ovo vrijedi u slučaju da se traženi podatak nalazi u RAM memoriji kontrolera, u slučaju da ga tamo nema, kontroler odmah izdaje instrukcije hard disku da pronađe i pročita podatke na određenoj lokaciji. Kod pisanja, računalo podatke umjesto na hard disk zapisuje u RAM memoriju kontrolera, da bi tek nakon toga kontroler počeo pisati po disku dok sam mikroprocesor računala već odavno zadovoljno obavlja svoje druge obaveze. Postoji mnogo načina na koje kontroler može pretpostavljati koje će podatke računalo zatražiti. Osnovne dvije metode su čitanje cijele trake hard diska ili čitanje samo sektora unaprijed. To znači da kada računalo zatraži podatak sa određenog mjesta na disku, kontroler će pročitati ili cijelu traku ili samo traženi i sljedeći sektor. Sve u svemu, princip hard disk keširanja kontrolera je vrlo sličan principu klasične keš memorije.

Prednost keširanja kontrolera se nalazi jed-



nim dijelom i u tome što njihova koncepcija „štedi" vaš hard disk, naime osim standardne „seek ignore" komande, kontroler sortire redoslijed čitanja ili pisanja, a osim toga većina podataka se nalazi u RAM memoriji kontrolera pa glava hard diska nemora konstantno putovati po površini.

...U ŠIROKO GRLO

Rad inteligentnog hard disk kontrolera počinje čim upalite računalo. Naime, zamislite da je vaše računalo file server za nekoliko desetina korisnika, da vaš kontroler upravlja sa desetak ogromnih hard diskova velikih kapaciteta. Kada upalite vaše računalo odjednom će svih desetak hard diskova iz vašeg nedužnog i nejakog mrežnog dijela računala zatražiti puno ampera struje, doći će do kompletnog kaosa. HyperSTORE u tom slučaju uključuje hard diskove jednog po jednog kako bi onemogućio strujni šok vašem računalu (vrijedi za hard diskove koji podržavaju tzv. „power on" komandu). Prilikom čitanja i određivanja koji će sektor kontroler pročitati i staviti u svoju memoriju hyperSTORE koristi preko dvadeset različitih faktora aktivnosti hard diska kako bi na najbolji mogući način heuristički algoritam donio odluku što spremiti u RAM kontrolera. Kod npr. pisanja podataka na disk hyperSTORE sve podatke pažljivo sortira u svojoj memoriji prema njihovoj poziciji na disku kako bi ih u što kraćem vremenu uspio zapisati na disk. Osim glavnog procesora (Z280) svaki mediadapter posjeduje svoj vlastiti procesor koji se bavi „low level" operacijama na hard disku poput čitanja, pisanja, pomicanja glave a sve zato da bi se glavni procesor paralelno sa pisanjem, čitanjem ili kojom drugom operacijom mogao baviti i komunikacijom sa računalom, dijagnostikom ispravnosti kontrolera.

Velika stavka na hyperSTORE kontrolerima je dana sigurnosti vaših podataka. Poput svih hard disk kontrolera hyperSTORE koristi ECC kodove za otkrivanje grešaka. HyperSTORE kontroleri za razliku od ostalih koriste 52-bitni ECC kako bi na najbolji mogući način osigurao podatke od njihovog oštećenja i gubitka. Kada kontroler naiđe na neki oštećeni sektor on ga pokuša nekoliko puta pročitati, ako ne uspije računalu šalje podatak o grešci. U slučaju da uspije, kontroler čita kompletnu traku, preformatira je i ponovno zapiše podatke na disk. U slučaju da pokušaj ne uspije, cijela traka se označava kao loša te se podaci sa nje premještaju na jednu od rezervnih traka. Ova metoda koje rezervira određeni prostor na disku za eventualne loše sektore uzima otprilike 1-1.5% prostora na disku, ali je zato disk praktički uvijek „error free" medij.

Za procesor Z280 je razvijen operativni sistem koji se svakodnevno usavršava. Kada kupite kontroler, u dva EPROM-a na samom kontroleru se nalazi najnovija verzija operativnog sistema nazvanog hSOS (hyperSTORE Operating System), kada na tržište izađe nova verzija operativnog sistema vi je jednostavno uz pomoć odgovarajućeg *utility* programa prenesete na hard disk, a kontroler umjesto iz EPROM-a kod svakog podizanja operativni sistem čita sa hard diska.

Operativni sistem hyperSTORE kontrolera osim svojih standardnih funkcija, može obavljati i razne dodatne funkcije poput „disk mirroring". *Disk mirroring* je opcija koja omogućava da iste podatke kontroler paralelno piše na dva različita hard diska. Ova metoda se koristi za automatski backup podataka na većim instalacijama. Osim toga, tu je i „drive spanning" koja omogućava da nekoliko fizički različitih hard diskova kompjuteru predstavimo kao jedan, te na taj način kontroler može baš kao i mainframe pristupati podacima uz pomoć „data striping" metode. Naravno, uza sve ostale mogućnosti hyperSTORE je sposoban operativnim sistemima poput DOS-a ili NOVELL-koji podržavaju diskove do najviše 1024 cilindra predstaviti hard disk koji ima više od 1024 cilindra kao hard disk sa manje od 1024 cilindra ali više glava, isto tako je moguće i jedan hard disk „podijeliti" tako da računalo misli da ima više fizičkih hard diskova. HyperSTORE kontroler je, za razliku od ostalih keširig kontrolera na tržištu, u mogućnosti upravljati i različitim optičkim sistemima.

ZAŠTO HYPERSTORE!?

Cijena jednog *hyperSTORE* kontrolera je vrlo visoka, veća čak i od cijene velikih hard diskova. Dok radite pod DOS-om, možda ćete reći da možete koristiti *Hyperdisk, Smartdrive* ili koji drugi keš program, i to je istina. Kupovati ovakav kontroler za mašinu za jednog korisnika je zaista šteta. Stvar se bitno mijenja ako ćete vaše računalo koristiti za velike CAD ili CAM projekte, ako ćete na vašem računalu pokretati velike baze podataka (npr. vođenje knjigovodstva velikih poduzeća), ili, kako to rade u Zagrebu, za BBS. Naime, zagrebački MIPS BBS je ovlašteni hyperSTORE support BBS za Evropu.

Prava snaga hyperSTORE kontrolera se otkriva tek na višekorisničkim sistemima. Možete si zamisliti kako se ponaša neki obični hard disk kontroler kada u jednom trenutku na hard disk navali desetak korisnika. Vjerujte da hyperSTORE tako nešto ni ne osjeti. Ako imate instaliranu Novell ili bilo koju drugu mrežu s puno korisnika, vrijeme pristupa podacima možete drastično smanjiti upotrebom hyperSTORE kontrolera. No, možda čak ni file serveri nisu toliko zanimljivi koliko je to npr. UNIX, koji konstantno nešto zapisuje po disku, traži, sortira, optimizira razno razne podatke. HyperSTORE kontroler se pokazao kao izuzetno dobro rješenje kod velikih *swap* datoteka, jer se praktički cijela datoteka nalazi u RAM memoriji kontrolera i time drastično ubrzava rad sistema. Za hyperSTORE kontrolere postoje driveri za sve operativne sisteme koje nalazite na PC računalima. Prednost ovog kontrolera je naspram ostalih keširig kontrolera da podržava interleave 1:1, što kod npr. svakako najoštrijeg konkurenta hyperSTORE-u, DPT SmartCache-a nije slučaj.

KONTROLER U AKCIJI

Kontroler hyperSTORE 1600 sam krenuo isprobati sa Imprimis Wren VI 94196-766 ESDI hard diskom (660MB formatirano), dok je model 400 bio zadužen za Maxtor LXTE200A, ▶

200MB IDE hard disk. Proces instalacije je vrlo jednostavan za svakog malo prosječnijeg korisnika, dok je čak i za apsolutnog početnika uz pomoć više nego ospežnog uputstva lagano shvatljiv.

U setapu sam saopštio računalu da hard diska više nema, nakon toga ga ugasio, otvorio kućište, izvadio iz njega ESDI Ultrastore kontroler, te sam na njegovo mjesto ubacio hyperSTORE 1600, spojio kablove, prekontrolirao dva puta i ponovo sve zatvorio. Odmah nakon toga sam sa *utility* diskete pokrenuo program pod nazivom „HS.EXE". Za ljubitelje eksplodirajućih prozora, ova instalacija će svakako biti doživljaj. Čak su i neki komentari poput „remember Murphy" ovdje da podsjete korisnika na višestruku opreznost. Odmah sam od raspoloživih opcija odabrao onu „Initialize physical drive". kontroler je odmah prijavio da se na prvom mediadapteru nalazi Wren VI ESDI disk, nakon pritiska na tipku enter, program je detaljno opisao ovaj hard disk sa svim njegovim parametrima. Jedna od najzanimljivijih je i MTBF vrijeme (vrijeme između kvarova) – hyperSTORE svakodnevno neumorno broji sate rada vašeg hard diska, kako bi vas na vrijeme upozorio da je vaš hard disk pred istekom vremena za koje proizvođač tvrdi da će raditi bez greške. Tu je i jedan parametar koji označuje broj staza koje ća kontroler rezervirati za svoj operativni sistem, različite dodatne opcije te za eventualnu relokaciju oštećenih traka.

Odmah nakon toga se pojavio „jelovnik" na kojem možete upisati sve loše sektore na disku, pokrenuti inicijalizaciju diska, sav proces zaustaviti ili promijeniti faktor analize diska. HyperSTORE prilikom formatiranja diska odmah provodi i jednu vrlo temeljitu analizu diska koja se označava stupnjevima od 0 do 9. Ako izaberete broj 9 tada je to najtemeljitija analiza diska i potrajat će možda čak i danima, dok će opcija 0 samo formatirati disk. Predložena brojka „2" je formatirala i analizirala disk pune 4 sata. Ovaj podatak možda izgleda velik, no svakako je bolje utrošiti 4 sata na više nego detaljnu analizu stanja površine vašeg hard diska, nego utrošiti mnoge sate i dane u pokušajima spasavanja uništenih podataka. Nakon formatiranja diska dolazi faza dijeljenja diska na logičke jedinice. Te logičke jedinice nisu iste kao i DOS-ove, nego njih računalo doživljava kao fizički odvojene diskove. Svoj Imprimis sam podijelio na dvije jednake logičke jedinice budući da DOS podržava diskove do maksimalno 1024 cilindra, dok Imprimis ima njih 1650. Odmah nakon definiranja logičkih jedinica dolazi odabir načina rada kontrolera. Ovaj korak je relativno kompleksan i usko povezan sa konfiguracijom logičkih jedinica, te zahtjeva dobro proučavanje uputstva za instalaciju.

HyperSTORE kontroleri podržavaju tri osnovna načina rada. Prvi način koji je podržan kod apsolutno svih operativnih sistema bez upotrebe drajvera je takozvani WD1003 kompatibilni mod koji je u novije vrijeme prozvan ISA standardom. Ako odaberete taj način rada sigurni ste da će vaš hard disk biti podržan od svih operativnih sistema, no morate znati da u tom načinu rada vaš hard disk radi najsporije, te da je veličina hard diska ograničena sa 1024 cilindra i 16 glava, s tim što možete imati maksimalno dva takva diska.

Drugi i treći način rada su vrlo slični. Prvi od njih se zove SSP *sector transfer mode*, dok se drugi zove SSP *block transfer mode*. Prednost ta dva načina rada nad ISA kompatibilnim modusom se sastoji u tome što je kontroler (navezan uz WD1003 kompatibilnost), u mogućnosti slati podatke računalu neusporedivo većom brzinom. To se postiže tako da se paralelno šalje puno više podataka nego što je to predviđeno ISA standardom. SSP *sector mode* šalje, kako samo ime kaže, po jedan sektor, dok SSP *block mode* šalje jedan blok.

U principu, ova dva načina rada su također podržavaju svi operativni sistemi, uz nekoliko izuzetaka. SSP načine rada mogu koristiti svi operativni sistemi bez upotrebe drajvera samo u slučaju da se sav I/O hard diska obavlja preko standardnog BIOS poziva 13H. Tu, dakako postoji ograničenje na dvije logičke jedinice (to je DOS, OS/2, PC-MOS i drugi). Uz upotrebu drajvera moguće je razbiti barijeru od dvije logičke jedinice, a uz to sam drajver povećava performanse kontrolera, tako da se npr. kod *Novell Netware* mogu paralelno prebacivati blokovi od čak 64KB. Operativni sistemi koji zahtjevaju drajvere su UNIX, XENIX, QNX i drugi multitasking operativni sistemi.

Odabirom načina rada, proces instalacije je završen te je ostatak prepušten operativnom sistemu tj. komandama FDISK i njima sličnima.

TO SE ZOVE BRZINA

HyperSTORE kontroleri model 400 i model 1600 se ponašaju identično, budući da su bazirani oko istog procesora i ista programske podrške. U svojem sedmodnevnom radu uz hyperSTORE kontrolere osjetio sam znatno ubrzanje sistema. Oba kontrolera su bila instalirana u njihovom „native SSP" načinu rada. Brzina kojom je ISC UNIX počeo obrađivati datoteke na disku bila je zapanjujuća. Dovoljno ilustrativan podatak je možda da je komprimiranje GNU C-a (ZOO arhiverom) uz pomoć Ultrastore ESDI kontrolera trajalo preko dva sata, dok je uz hyperSTORE posao bio obavljen za nešto manje od petnaest minuta. Kod DOS-a je ubrzanje bilo više nego fascinantno, osim što je *Coretest* uporno javljao „results probably due caching". Rezultati su bili zapanjujući: 3.6MB u sekundi, uz vrijeme pristupa od 0.3 milisekunde. Na žalost nisu mi bili dostupni drugi brzinski testovi, no rad uz Venturu je postao neusporedivo brži, a sortiranja mojih sitnih baza podataka je postalo pitanje desetinki sekunde. Sortiranje podataka na 300MB DOS particiji uz program COMPRESS iz paketa *PcTools* trajalo je nešto više od 6 minuta.

HyperSTORE kontroleri su na meni ostavili više nego impresivan dojam. HyperSTORE na tržištu nije sam. Dapače, uz njega je još mnogo konkurenata, no u gotovo svim testovima, od *PcMagazin*-a pa sve do *BYTE*-a, hyperSTORE je dobio najbolje moguće kritike, čak je i od strane uredništva BYTE-a odlikovan sa nagradom „Award of tehnical excellence" – što zasigurno govori o njihovoj kvaliteti. HyperSTORE kontroler model 400 bih preporučio bilo kojoj mreži koje nema namjeru rasti u beskonačnost. HyperSTORE 1600 bi ostavio svakom ozbiljnom sistemu koji se ima namjeru vremenom proširivati i rasti, jer osim samih superlativa u pogledu same keširng tehnologije, hyperSTORE 1600 je i jedini kontroler koji podržava kapacitet od 50.4 gigabajta. Ovaj članak bih htio završiti rečenicom iz BYTE-a koja glasi „Keširanje je komplicirana tehnologija, a ljudi u PSI-ju izgleda o tome znaju više od bilo kojeg konkurenta".

Kontakt adresa

CC&T
I. Luke 11, 41040 Zagreb
Tel: 041/269-525, fax: 041/265-410

"Kneza Miloša Strasse"

PRVA PRAVA RAČUNARSKA PRODAVNICA

Oči su ogledalo duše, a izlog – ogledalo prodavnice. Jedan izlog u ulici Kneza Miloša 4, na manje od sto metara nizbrdo od Savezne skupštine (šta li to beše?), mada će ostaviti ravnodušnim mnoge užurbane prolaznike, dirnuće u srce svakog poklonika Boga računarstva. Taj neveliki izlog „Jugodata" i „Univerzala" će računarske „veterane" neodoljivo podsetiti na Schillerstrasse (Šilerštrase) u Minhenu, čuvenu ulicu u kojoj se nalaze praktično sve glavne prodavnice računarske opreme (otuda igra reči u našem nadnaslovu!). Za računarske „novajlije" to će biti scena kakvu im „duša ište": u izlogu svega – od kompletnog AT-a (originalni IBM) i pojedinačnih komponenata, preko softverskih paketa i literature, do papira i drugih potrošnih materijala.

„Jugodatu" pamtimo po akcijama „Singapurska veza" i „$799 – najbolja cena na svetu". U međuvremenu, ona je proširivala ponudu novim uređajima, komponentama, softverom, uslugama. Danas je to jedno uspešno preduzeće, sa lepim poslovnim prostorom u Save Kovačevića 6, razvijenim kontaktima sa mnogim uglednim stranim i domaćim firmama i sa poslovnom filozofijom da, kako kažu, „kupcu za isti novac uvek treba dati još nešto (pomoć, obuku, vlastiti softver)".

Među to „još nešto" spada i prva prava specijalizovana prodavnica računarske opreme u Beogradu (a, koliko znamo, i „šire" u našoj zemlji). Prodavnica je vlasništvo beogradskog „Univerzala", koji je dao i potrebnu opremu prodavnice i organizovao službu prodaje. Računarsku opremu i sve ostalo obezbeđuje „Jugodata". U prodavnici se mogu dobiti računari IBM i Litron, laserski štampači Hewlett Packard, štampači Epson, Fuji, Oki, pojedinačne komponente, monitori, originalni softver vodećih svetskih firmi (Microsoft, Borland, Nor-
ton), kablovi, navlake, držači za papir, papir i upijače za štampače, sredstva za čišćenje, tu je i domaća računarska literatura, a u planu je i nabavka specifične, namenske strane literature. Dakle, za svakog ponešto – a za mnoge i sve!

Ali ono što – da se vratimo na prvi utisak – mora da dirne u srce svakog PC sladokusca, to je izlog u kome možete da vidite praktično sve što vam nudi ova prava računarska prodavnica.

E.J.

dvobroj za juli i avgust časopisa

računari

izlazi iz štampe 15. jula 1991.

SAMOLEPLJIVI NATPISI

STIKA – *Intelligant Cutting Machina*. Iza ovog imena se krije uređaj sa kojim većina čitalaca zasigurno nije imala prilika da se sretne. Zato je najbolje na samom početku objasniti: Šta je STIKA? Skaner i rezač u jednom pakovanju.

Pavle Peković



Ako volite samolepljive natpise. Skener i rezač u jednom pakovanju

STIKA je mašina koja služi za isecanje željenih oblika na samolepljivoj hartiji. Mustra na osnovu koje želimo da isečemo hartiju (logo znak, slova...) se prenese u memoriju računara iz koje, kada zaželimo, možemo iseći zapamćeni oblik. Obe ove funkcije – skeniranje uzorka i njegovo isecanje objedinjene su u jednom uređaju.

U MALOM PAKETU

STIKA radi na 16V i sa njom smo dobili ispravljač od 14.5V. Zanimljivo je da je na traci koja je bila okačena na kabal pisalo 16V. Ipak, ova sitnica nije smetala STIKA-i da narednih dana radi pouzdano i bez ikakvih problema. Ispravljač se podnošljivo grejao i nismo primetili nikakve smetnje. Ispravljač je proizveden na Taiwan-u ali, što je daleko važnije, STIKA dolazi iz Japana i delo je čuvenog proizvođača računarske opreme, Roland-a.

U paketu smo još našli i rezervno sečivo koji ćete staviti kada staro izgubi oštrinu. Naravno, rezervna sečiva, kao i sav detalj potrebni pribor ćete dobiti kod preduzeća MZ iz Beograda, čijom smo ljubaznošću i dobili STIKA-u na testiranje. U paketu su još i dve knjige. Jedna je priručnik za upotrebu, kojem možemo zamerili jedino nepostojanje preciznijih informacija o izlazu za računar. Druga knjiga je tzv. „knjiga ideja" u kojoj se nalaze razne sličice, slova i drugi često korišćeni oblici koje možete praktično upotrebiti kao uzore prilikom isecanja nalepnica i hartija. Naravno, u paketu se nalazi i nekoliko uzoraka papira, traka i nalepnica, kao i folija koja je namenjena skeniranju sa neravnih površina.

Sama STIKA se nalazi u četvrtastom plastičnom kućištu, sive ili prljavo bele boje, po želji kupca, čije su dimenzije 202×146×80 mm, a težina je 1190 g.

Uključivanje uređaja je proteklo bez problema. Kada se STIKA uključi, glava za rezanje se centrira kao kod matričnih štampača, što izaziva prilično neobičan zvuk, koji deluje „metalno" i uliva uverenje u robusnost uređaja.

SKENIRANJE...

Prva operacija, koju valja uraditi, je skeniranje slike koju želimo iseći na hartiji. Skener koji je ugrađen u uređaj je ručnog tipa. Nalazi se sa donje strane kućišta i radi tako što se ceo uređaj pomera preko željenog crteža. Tome pomaže ugrađeni valjak sa donjeg dela skenera koji omogućuje lakše pomeranje po površini, kao i prozorčić kroz koji se može videti deo crteža koji se skenira. Za vreme skeniranja se uključuje i lampica, koja osvetljava crtež ispod. Dakle, na prvi pogled, sve je kao kod ručnih skenera.

Da biste preneli crtež u memoriju, potrebno je podesiti još neke parametre. Pre svega, to je širina skenirane površine. Skener omogućuje četiri različite širine skeniranja, od kojih zavisi i maksimalna dužina skeniranja.

širina		dužina
64 mm	x	120 mm
48 mm	x	190 mm
32 mm	x	240 mm
16 mm	x	240 mm

Nakon toga, potrebno je podesiti preciznost skeniranja. Moguće su dva nivoa preciznosti, tzv. NORM sa 200 dpi (tačaka po inču) što je oko 8 tačaka po milimetru i tzv. ZOOM sa 400 dpi ili oko 16 tačaka po milimetru. Ostaje još da podesimo osvetljenost crteža i možemo početi sa skeniranjem. Skenirati se može u dva pravca sa različitim efektima. Ako se STIKA vuče unapred, izrezane slike će biti identična skeniranoj. Ali, ako se skenira u suprotnom pravcu, tj. ako se skener vuče unazad, izrezana slika će biti slika u ogledalu skenirane slike.

Ovo je, međutim, samo početak iznenađenja. Postavljali smo mu sve i svašta da skenira i uređaj je se svim izlazio na kraj bez ikakvih teškoća – bez ikakvih „mrlja" prenosio je čak i lošije crno bele crteže, ali se nije bunio ni kada smo mu podmetnuli crteža u crvenoj, plavoj ili nekoj drugoj boji. Jednostavno, radio je bez greške. Naravno, bilo bi previše očekivati da skenira svetlo-žuta slova na beloj pozadini ili pak tamnije slova na nekoj tamnijoj pozadini, ali smo i pri takvim pokušajima dobijali rezultate sa kojih se ponekad mogao prepoznati skenirani crtež.

Skeniranje se vrši pritiskom na najuočljiviji i najveći taster na uređaju. Moramo samo uočiti da je taster postavljen u položaj koji ne odgovara levorukima. Takođe, moramo pomenuti da je zbog veličine samog uređaja potrebno malo više mesta na ploči na kojoj se vrši skeniranje. Držanje pravca pri skeniranju je veoma lako, a promene u brzini skeniranja, naravno ako odstupanja nisu velika (manje od 15 mm/sec), ne utiču na kvalitet skenirane slike.

...I REZANJE

Na novom uređaju je potrebno prvo postaviti sečivo. Za to je neophodno otvoriti četvrtasti plastični poklopac koji skriva mehanizam za rezanje. Sečivo je odmah ispod poklopca, a sama montaža je krajnje jednostavna. Sečivo se nalazi na horizontalno postavljenom nosaču koji omogućuje njegovo pomeranje po x osi. Sečenje po y osi se ostvaruje pomeranjem papira, za šta su zadužena dva točkića koji pomeraju papir napred i nazad. Točkići veoma čvrsto drže papir, što na njemu ostavlja vidljiv tačkasti trag. Stiče se utisak da bi bilo korisno da je ugrađen i drugi par točkića sa druge strane ose sečiva, jer bi se tako papir bolje zalezao. Ovako nam se par puta desilo da STIKA zgužva samolepljivu hartiju. Takođe, možda bi bilo bolje kada bi se umesto točkića koristili valjci, jer bi time bilo omogućeno korišćenje i uže hartije.

Korisna adresa

MZ
Dubljanska 70, 11000 Beograd
Tel. (011) 434-612 Fax: (011) 450-471

Sada je moguće koristiti samo hartiju širine oko 65 mm. Dužina hartije se kreće između 50 i 600 mm. Ipak, površina rezanja je nešto manja od dimenzije papira. Širina je smanjena na nešto iznad 60 mm zbog točkića koji guraju hartiju, pa se ovde još jednom potvrđuje da bi bila bolja ideja sa valjcima umesto točkića. Po dužini, prvih 23 mm i zadnjih 7 mm STIKA ne koristi. Takođe, treba voditi računa o debljini papira koja mora iznositi 0.16 mm da bi STIKA pravilno radila.

Pre rezanja potrebno je podesiti veličinu izrezane slike u odnosu na skeniranu sliku. Moguće je preneti sliku identičnu skeniranoj, kao i uvećanu dva ili tri puta. Opcijom FULL, skenirana slika se modifikuje tako da maksimalno iskoristi veličinu papira. Uvećanje/umanjenje slike je moguće od 0.625 do 4 puta.

Rezanje počinje umetanjem papira u prostor za rezanje. Uređaj najpre uvlači ceo papir da bi mu izmerio dužinu. Deklarisana brzina rezanja je 20 mm/s, s tim što je realna brzina manja za vreme koje odlazi na dizanje i spuštanje sečiva, kao i na pomeranje papira i sečiva.

Treba skrenuti pažnju na to da je potrebno koristiti što kvalitetniji papir propisane debljine. Najbolje je koristiti originalni papir koga možete nabaviti kod preduzeća MZ. Time se smanjuje rizik gužvanja papira, a i preciznost sečenja je veća. Pri tom ne treba previše očekivati od skeniranja i sečenja jako sitnih detalja. Naime, najmanja slova koja smo uspeli da skeniramo/uvećamo/izrežemo bila su bile veličine 9 tačaka.

SVE I SVAŠTA

U uređaj je ugrađen i izlaz za računar. Pravo je čudo što proizvođač nije više mesta posvetio ovoj mogućnosti, jer to širi spektar primene uređaja. Nešto više informacija smo dobili od preduzeća MZ, gde su nam rekli da se na izlazu dobijaju podaci u ploter formatu koji, naravno, razumeju svi Rolandovi ploteri. Raspored pinova na konektoru i sam izgled konektora do sada nismo videli, dakle nije standardan, što nagoveštava da će Roland, ako već nije, početi prodavati potreban kabel i, ako je potrebno, softver, po nekoj nerealno velikoj ceni. Videli smo ovakve slučajeve dosta puta u praksi.

Na kraju treba pomenuti šta sve korišćenjem ovog uređaja možemo dobiti. Isecajući samolepljivu hartiju, možemo praviti nalepnice najrazličitijih oblika, ali i samolepljive natpise kojima možemo obeležavati, recimo, diskete, video kasete, ili, kako su to lepo nacrtali u „knjizi ideja", natpise za tegle, pisma, fascikle... Uz pomoć skeniranja kojim se dobije slike u ogledalu, moguće je iseći nalepnice koje se lepe sa unutrašnje strane stakla a čitaju spolja. Za mlade, najzanimljivije će sigurno biti isecanje materijala koji je moguće kućnom peglom zalepiti na delove odeće.

Dakle, spektar primene je zaista širok, i sigurni smo da će STIKA osvojiti mnoge, prvo svojom praktičnošću a zatim i svojim kvalitetom. Niko naravno nije bez mane, pa ni ovaj uređaj, ali njegove prednosti su daleko veće.

SVIRKATA NA PISIITE

Sigurno ste bili u prilici da čujete zvučne mogućnosti računara kso što su amiga, atari ili mekintoš. Verujemo i ds niste baš oduševljeni zvučnim potencijalima vašeg PC-a. Rešenje postoji: zvučna kartica sa zvučnim imenom SoundBlaster.

Siniša Mihailović

Zahvaljujući svojoj orjentaciji na poslovno tržište, PC računari su, što se tiče zvučnih potencijala, jako siromašni. Prošla su vremena kada je zvuk na računaru bio potreban samo da nas upozori ili da nešto signalizira. Na većini PC-kompatibilnih računara zvuk, najčešće, proizvodi beeper. Pojavom novih trendova, kao što je multimedia, koji zahtevaju dobar zvuk, ovo počinje da se iskazuje kao nedostatak. Mnogi klinci, koji u svet računara ulaze preko računarskih igara, čežnjivo gledaju igrice sa neverovatnim zvučnim efektima, na ekranima svojih drugova koji imaju amigu, atari ili nešto drugo. Zvuk, međutim, može da ima i veoma ozbiljnu primenu.

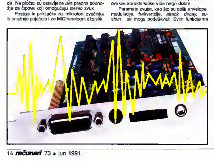
Uz poruku „Blast sway the PC sound barrier" (probijte zvučni zid PC-a), hardver i softver koji doslovno sprovode ovaj slogan, kompanija Creative Technology, lansiranjem kartice SoundBlaster, nastoji da postavi novi standard za sintezu zvuke na PC kompatibilnim računarima.

ZVUČNA KARTICA

Kartica SoundBlaster radi na svim PC/XT AT, 386, PS/2 (model 25/30) i Tandy (osim 1000EX/HX) mašinama a zahteva jedan osmobitni slot. Štampana pločica deluje solidno – opšti utisak kvari jedino otpornik zalemljen preko jednog od čipova.

Pločicom dominiraju, svojom veličinom, dva čipa sa oznakama proizvođača ove kartice – Creative Technology. Jedan od njih predstavlja integrisani FM generator, a drugi je A/D – D/A konverter. Pored njih, na pločici se nalazi i dvadesetak TTL čipova, tajmer 555 i par namenskih čipova. Tu su i 4 džampera za izbor I/O adrese (210 – 260H), interapt linije (IRQ 2 3 5 ili 7) i isključivanje džojstika i DMA moda. Na pločici su ostavljene dva prazna podnožja za čipove koji omogućuju stereo zvuk.

Postoje tri priključka za. mikrofon, zvučnika ili snažnije pojačalo i za MIDI/analogni džojstik.



Na kartici su smešteni:

- FM generator zvuka (AdLib kompatibilne muzička kartica)
- D/A konverter za reprodukovanje digitalizovanih zvukova
- A/D konverter za digitalizovanje zvučnih signala
- Džojstik port
- MIDI interfejs
- Stereo pojačalo 4W, 4 oma, priključak za zvučnike/dodatno pojačalo
- mikrofonsko pojačalo i priključak za mikrofon

a opciono se mogu nabaviti dva čipa koji zvučnim sposobnostima ove kartice, pored još jednog kanala (ukupno 12), dodaju novu dimenziju – stereo zvuk.

GENERATOR ZVUKA

U reklami piše da je SoundBlaster u osnovnoj konfiguraciji u stanju da proizvede maksimalno jedanaest različitih FM zvukova simultano. Naravno, ovo se odnosi na FM generator SoundBblastera. Zvuči jako lepo, međutim u praksi je malo drugačije.

SoundBblasterov FM generator konsti 18 „ćelija" za operatore. Za sintezu jednog zvuka su potrebne dve ćelije „modulator" i „nosač". Prostom računicom se dolazi do zaključka da je maksimum 9 zvukova. Tih 18 ćelija se može upotrebiti na tri načina. Jedan je već spomenut, drugi je sinteza govora a treći, u kome se prvih 12 ćelija koriste za 6 kanala FM zvuka a ćelije 13-18 za 5 udaračkih instrumenata, je iskorišćen za reklamu. Međutim, jasno je da su i ovakve karakteristike više nego dobre.

Parametri zvuka, kao što su oblik envelope, modulacija, frekvencija, *attack, decay, sustain*, se mogu podešavati. Svim funkcijama FM generatora upravljaju podaci, upisani u registre.

Za sintezu govora se brine drajver SBTALKER. Ovaj drajver je prilagođen engleskom govornom području. Nažalost, u priloženoj dokumentaciji nigde nema podataka o mogućnosti prilagođavanja drajvera drugim govornim područjima. Bez odgovarajućeg drajvera potrebne su razne egzibicije da bismo naterali karticu da „progovori" na našem jeziku. Tako, na primer, da bismo proizveli reč računari, tekst koji prosleđujemo kartici izgleda kao „rachoonaeree".

Jasno je da je „čitanje" našeg teksta iz datoteka, što bi se moglo dobro iskoristiti za „ozvučene" prezentacije, praktično nemoguće.

A/D I D/A KONVERTER

A/D konverter služi za pretvaranje analognih u digitalne signale, u ovom slučaju za digitalizovanje zvuka. Zvučni signal se sempluje (uzorkuje) tako što se, u jedinici vremena, određen broj puta uzima njegova vrednost. Što je broj uzorkovanja po vremenskoj jedinici veći, digitalizacija je vernija originalu. SoundBblasterov A/D konverter sempluje frekvencijom 5 – 12 kHz.

8-bitni D/A konverter obezbeđuje reprodukovanje digitalizovanog zvuka. Frekvencija semplovanja je programabilna i kreće se od 4 do 23 hHz.

A/D i D/A konverter rade u dva transfer moda:

- direktni, procesor prenosi bajt po bajt sempla
- DMA, nije potrebna intervencija/nadzor procesora

Digitalizovan zvuk je pravi potrošač memorije. Na primer, sempl koji je pravljen na frekvenciji 8 kHz, od zvuka u trajanju od tridesetak sekundi, može zauzati preko 300Kb. Sa povećanjem frekvencije semplovanja (A/D konverter omogućava frekvenciju semplovanja do 12kHz), dužina sempla, za istu vremensku jedinicu, drastično raste. Zbog ovoga je nekakav način pakovanja semplova nužan. Semplove je sa 8 bita moguće komprimovati na:

- 4 bita (dva prema jedan), gubitak na kvalitetu je neznatan
- 2 6 bita (tri prema jedan), gubitak na kvalitetu se oseća
- 2 bita (četiri prema jedan), gubitak na kvalitetu je značajan

Dekompresija pakovanih semplova je hardverska. Pored ovog moguće je vršiti i zamenu perioda tišine sa po samo par bajtova, koji označavaju dužinu perioda tišine. Prag tišine se takođe može podesiti. Gubitaka na kvalitetu nema.

MIDI INTERFEJS

MIDI interfejs omogućava povezivanje sa muzičkom opramom koja
sadrži ovaj interfejs kao što su sintesajzeri, ritam mašine, efektori, itd. Da bi konstili MIDI

interfejs morate dokupiti adapter koji ima izvedene kablove za MIDI in i MIDI out, kao i priključak za džojstik. Ponuđen je i MIDI *connector box* koji ima pet MIDI out izlaza, jedan MIDI in ulaz i priključak za džojstik.



SOFTVER I PODRŠKA

Uz karticu se dobijaju 4, 1.2M i/ili 2, 1.44M diskete. Na prvoj disketi se nalazi instalacioni program, koji na hard-disk kopira sve programe i drajvere.

Pri instalaciji se mogu javiti problemi, zbog prisustva drugih kartica koje koriste istu I/O adresu i/ili interapt liniju kao i SoundBblaster. Problem se otklanja jednostavno, promenom adrese ili IRQ linije, na džamperu. Ovaj kao i drugi mogući problemi i rešenja su detaljno obrađeni u uputstvu, koje se dobija uz karticu.

U uputstvu uz karticu je dat i opis programa koji se dobijaju, uz nju.

Voxkit

Disketa sa ovim nazivom sadrži četiri programa za snimanje, reprodukovanje i ograničenu manipulaciju sa semplovima. Program koji nosi naziv *VOXKIT* omogućava snimanje semplova u memoriju ili direktno na disk, reprodukovanje sa diska ili iz memorije i pakovanje semplova. Programi *VREC*, *VPLAY* i *VJOIN* služe za snimanje, reprodukovanje i spajanje semplova iz DOS-a.

FM intelligent organ

Namena ove zbirke programa je kreiranje i reprodukovanje muzike. Program *FMORGAN* omogućava pravljenje jednostavnih pesmica služeći se fondom instrumenata FM generatora zvuka. Nažalost nikakve mogućnosti editovanja nema, unos je sa tastature ili sa klavijature (vezane preko MIDI interfejsa). Ovaj program je, po rečima proizvođača, dizajniran za ljude koji ne poseduju muzičko obrazovanje. Po mogućnostima jako podseća na igračke tipa CASIO digitron/klavijature, s izuzetkom što *FMORGAN* ima i mogućnost snimanja/učitavanja podataka. Kao i kod pomenutih, Casio klavijatura, i ovde važi princip „one finger play" (sviranje jednim prstom) znači monofono, uz „inteligentnu" automatsku pratnju i ritmove. Program *FMPLAY* služi za reprodukovanje FM muzike iz DOS-a. Pošto prenos podataka ide preko DMA, za vreme dok svira muzika, moguće je pokrenuti neki program ili izvršiti DOS komandu, što se, kao parametar, prosleđuje programu *FMPLAY*.

Sbtalker with DR Sbaitso

Sbtalker je program koji omogućava izgovor pojedine reči ili čitavog teksta (iz ASCII datoteke), naravno na engleskom jeziku, koristeći FM generator zvuka.

DR Sbaitso je program koji će vas, ukoliko dobro vladate engleskim jezikom, zabaviti. Pred vama se odjednom pojavljuje računar/psihijatar koji vas, umornim glasom, poziva da mu izložite svoje probleme. Ovaj program raspolaže dosta velikim fondom reči, tako da se može razviti zanimljiv „dijalog" uz duhovite komentare računara.

Talking parrot

Kao što samo ime kaže, papagaj koji govori. Ponavlja skoro sve što stigne „do njega" kroz mikrofon, uz povremene komentare tipa „Auh, al' imaš loš zadah", „Ništa te ne čujem", „Šta mumlaš?", itd.

Vedit (Creative technology)

Ovaj program služi za manipulaciju sa digitalizovanim zvukom. Pored funkcije snimanja i reprodukovanja zvuka, na željenoj frekvenciji kao i pakovanja semplova, koje su, inače, uključene u program *VOXKIT*, ovim programom se mogu editovati semplovi. Sempl koji se edituje je moguće zumirati i deliti na blokove, dodavati mu eho.

Prikaz sempla je dat grafički, što u *VOXKITU* nije slučaj. Priručnik je tanak ali su sve opcije dobro objašnjene. Ovaj program radi samo sa EGA/VGA karticom.

Spellbox (Creative technology)

Spellbox je edukativni program. Služi učenju slova i izgovora. Ovde ne postoji problem sa govornim područjem pošto se koristi digitalizovani zvuk. Rad sa ovim programom je sličan magnetofonu. Postoje već uanimljeni glasovi ali postoji i mogućnost pravljenja novih. Posebno je interesantno što se u lekcije može uključiti animacije.

Sekvencer SP junior (Voyetra)

Naziv sekvencer je nastao iz reči sekvenca. Ova kategorija programa omogućuje „snimanje" muzike, najčešće preko MIDI interfejsa. Reč snimanje je stavljena pod navodnike jer se ne radi o snimanju tonova već brojeva koji predstavljaju informacije o muzičkim tonovima. Muzičari ove programe, najčešće, koriste za izradu muzičke pratnje, koje se sastoji od sekvenci koje se ponavljaju. Naravno, sa snimljenim informacijama se može manipulisati (mogu se menjati, premeštati, brisati...) i mogu se reprodukovati kao muzika.

Sekvencer programerske kuće Voyetra „Seqencer Plus Mark III", dva puta je izabran kao najbolji od strane urednika časopisa „PC magazine".

SP junior je sekvencer donjeg cenovnog razreda i svoju reputaciju, po rečima proizvođača gradi na osnovu slave „starijeg brata". Ovaj sekvencer, doduše, omogućuje rad i sa FM generatorom SoundBblastera i sa MIDI interfejsom, kao i mešanje zvuka FM generatora i MIDI klavijature, ali ne postoji mogućnost notnog editovanja (**score edit**), dati su interni limiti za snimanje, dokumentacija je oskudna, nisu dobro opisani problemi koji se mogu javiti (na primer, za vreme reprodukcije ne čuje se ton na klavijaturama, iako je sve dobro spojeno, itd.)

MMPLAY (Creative labs)

MMPLAY je program koji objedinjuje animaciju *Autodesk Animatora*, FM i digitalizovan zvuk u jednu celinu. Program radi sa *batch* datotekama, u koje su uključena komande, koje podržava. Zahvaljujući tome što *Autodesk Animator* postavlja markere u animaciju, a isto to čine i programi koji podržavaju zvučne sposobnosti SoundBblastera, moguće je lako sinhronizovati animaciju, digitalizovan zvuk i muziku. Ovaj program je namenjen „ozvučenim" prezentacijama.

Korisna adresa

Smilex d.o.o
Vojvode Dobrnjca 8, Beograd
Tel: 011/767-516, 750-197
Fax: 011/750-197

Razvojni alat

Developers kit je namenjen svima koji žele da razvijaju programe koji podržavaju SoundBblaster. Ovde su date biblioteke funkcija i procedura, namenjenih radu sa karticom, za Microsoft C i *Quick Basic*. Na prvi pogled ovaj paket izgleda dovršeno, u priručniku od par stotina strana ima dosta podataka i primera, međutim, neke stvari, kao što je na primer struktura *Sbtalker* drajvera su preskočene ili šturo objašnjene.

PRIMENA U PRAKSI

Da bi jedna kartica uspela na tržištu, uslov je da ima dobru softversku podršku. Najveći deo softverske podrške za SoundBlaster čine igre. Praktično više od polovine. Ali ima i „ozbiljnog" softvera koji podržava SoundBblaster i proizvođača kao što su IBM (*Story Board Live*), Microsoft (*Windows 3.0*), *Voyetra* (poznata po sekvencerima), Lotus (*Lotus notes*), itd. Nepotpuna lista, dobijena od zastupnika, sadrži preko 200 naslova.

Jedan broj ovih programa je, za potrebe ovog prikaza, testiran i praktično primenjen za vreme Sajma tehnike, na štandu „Sezama" SoundBblaster je „govorio" glasom Zorana Modlija, dok se na ekranu „vrtela" animacija. Čitav posao oko prezentacije je odneo vreme za pravljenje animacije, 20-tak minuta za pripremu muzike, 30-tak minuta za digitalizaciju dvominutne poruke Zorana Modlija i sinhronizacije sa animacijom. Dva minuta digitalizovanog zvuka, na 8 kHz, zauzelo je 1.2 MB na disku.

Interesantno je pomenuti da je reklamna poruka digitalizovana na samom štandu, usred buke koja je vladala, ali zahvaljujući mogućnosti silence pakovanja sav pozadinski šum je potisnut. U ovom poslu je mnogo pomogao program *VEDIT*. Muzika je obrađena sa programom *SP junior*, a sve je objedinio program *MMPLAY*.

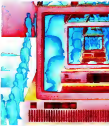
Pažnju posetilaca su privukle i neizbežne igre, koje podržavaju ovu karticu: *Where in the world is Carmen San Diego*, *Stellar7* i popularni *Prince of Persia*.

U reklami za karticu stoji „The PC sound standard". SoundBblaster nudi širok spektar mogućnosti. Kompatibilan je sa Adlib muzičkom karticom, što donosi dodatnu softversku podršku. Imajući u vidu cenu, mogućnosti i softversku podršku, na Zapadu bi ova kartica stvarno mogla da postane novi standard za zvuk na PC-u. Na istočnom tržištu, cena od US $240 ne deluje baš privlačno.

BRZINE NIKAD DOSTA

U Beogradu je od 5-9. maja održan seminar pod nazivom „Moderne računarska arhitakture" u organizaciji Sava Centra, Elektrotehničkog fakulteta u Beogradu i Instituta Mihailo Pupin. Malobrojni učesnici imali su sreću da kroz predavanja i neformalna razgovore sa domačim i stranim predavačima otkriju neke od mogučih pravaca daljag razvoja računarskih arhitektura.

Aleksandar Gobović



Težište seminara postavljeno je na što širi pregled dosadašnjih tehnika i probleme na koja se nailazilo, sa ciljem da se trasiraju nova rešenja. Ovaj očigledno edukativni i razvojni zadatak potpomogla je NCR korporacija koja u saradnji sa ekipom sa ETF-a i IMP a okupljenom oko profesora **Veljka Milutinovića**, pokušava da dizajnira novu generaciju računara sposobnu da odgovori na narastajuće zahteve multimedijalnih aplikacija.

SVE VEĆI ZAHTEVI

Moderni termin „multimedija" već je poznat pratiocima vesti koje stižu sa Sezama. Ukratko, on obuhvata sposobnost računara da se nosi sa „prozorskim okruženjem" (windowing environment), fotorealističnim prikazom, animacijom, kvalitetnim zvukom, pretraživanjem velikih količina informacija i sve to u interaktivnom obliku. Kvalitetno obavljanje pomenutih zadataka tvrd je orah i za nove generacije radnih stanica, a kamoli za PC računare. Prvi multimedia standardi koji se zadovoljavaju i286 procesorom već postoje, mada uz ogromne kompromise. Međutim, očekivani dolazak i586 procesora i sve češće korišćenje RISC procesora kao ubrzivača daje nadu da će se i na našim PC zavrteti aplikacioni softver devedesetih, multimedia.

Da bismo shvatili „tešku artiljeriju" kojom se napada večni problem nedovoljno brzog računara, razjasnimo osnovne termine koji su se koristili tokom seminara. Pretpostavimo da obavljanje jedne instrukcije ima četiri faze: IF – dohvatanje instrukcije iz memorije (*instruction fetch*), ID – dekodiranje instrukcije (*instruction decode*), EX – izvršavanje (*execute*) i ME – obraćanje memoriji. Ovaj slad faza se kod tradicionalnih računara ponavlja jedan za drugim. Prvo ubrzanje nastalo je kada se uvidelo da se obavljanje jedne faze instrukcije može vršiti paralelno sa obavljanjem drugih faza. To je dovelo po protočne obrade (*pipeline*). Osnovu svakog RISC procesora danas čini neka *pipeline* varijanta. Usitnjavanje instrukcija na veći broj manjih faze ograničen je unutrašnjim paralelizmom procesorskih resursa. Manje, odnosno kraće faze, omogućavaju veću frekvenciju takta, jer je perioda takta jednaka vremenu izvršavanja jedne faze. Cilj ovako zamišljene mašine je postizanje izvršavanja jedne instrukcije po taktu, ili recipročno jednog takta po instrukciji (*cycle per instruction* – CPI). Taj cilj se u realnim uslovima ne može ostvariti ovakvom arhitekturom zbog narušavanja protočnosti prilikom prekida i grananja. Da se ode ispod CPI=1 potrebni su „super" prefiksi.

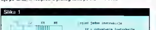
Superskalarni računari istovremeno očitavaju nekoliko instrukcija i paralelno ih obavljaju. Za to je potrebno imati više izvršnih jedinica (kao što su jedinice za celobrojnu i aritmetiku u pokretnom zarezu) unutar procesora, sposobne za paralelan rad unutar jedne faze instrukcije. Superpajplan (*superpipeline*) organizacija nema za direktan cilj smanjenje CPI, već smanjivanje vremena mašinskog ciklusa. To se postiže internim preklapanjem ciklusa. Mašine sa vrlo dugom instrukcijskom reči (*very long instruction word* – VLIW) čija instrukcijska reč ima najčešće 128-512 bita, često se koriste kao matematički ubrzivači. Vektorska mašine istovremeno obrađuju nizove brojeva jednom instrukcijom i postižu odlične performanse u numeričkim primenama. Uticaj koncepta vektorske mašine dominantan je kod CRAY 1 superračunara.

POBEDNIK JE POZNAT

Dilema RISC ili CISC danas više ne postoji. Podsetimo se zato osnovnih odlika pobednika RISC arhitektura. To je pre svega **Load / Store** koncept koji forsira obavljanje svih operacija isključivo nad procesorskim registrima i odvaja ih od sporih instrukcija obraćanja memoriji. Zatim, tu je osobina po kojoj je RISC (*Reduced Instruction Set Computer*) dobio ime: pažljivo odabran skup instrukcija, koji je u početku bio znatno manji od CISC (*Complex Instruction Set Computer*) procesora, ali se danas ta razlika ne primećuje. Razlog tome je po rečima **Veljka Oklobdžije** iz IBM T.J. Watson Research centra u pažljivom odabiru instrukcija koje će biti implementirane u hardveru procesora. Ukoliko dodavanje novih instrukcije poboljšava performanse procesora i ne kvari performanse već postojećih, onda nema racionalnog razloga da ukupan broj instrukcija implementiran na nivou hardvera bude mali. Iz prve osobine sledi i treća: jednostavni načini adresiranja. Sve instrukcije su iste veličine i fiksnog formata, što olakšava dekodovanje. Protočna obrada je obavezna i nameće cilj od CPI=1. Zbog naredbi grananja koje narušavaju protočnost, potrebna je čvrsta sprega kompajlera sa hardverom računara kako bi se optimizovalo iskorišćenje računara u takvim situacijama.

Razvoj tehnologije ima presudan značaj za

Slika 1



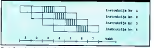
Mašinska instrukcija i njene faze

Slika 2



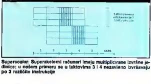
Sekvencijalno izvršavanje instrukcija: Tradicionalna računarska arhitektura obavija instrukcija jednu za drugom; izvršne faze su osenčene

Slika 3



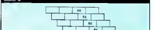
Protočne obrada – *pipeline*: Kod protočnosti se sve faze instrukcije mogu obaviti u jednom taktu, u ovom primeru se pune protočnost postiže tek u četvrtom taktu

Slika 4



Superscaler: Superskelemi računari imeju multiplicirane izvršne jedinice; u nešem primeru se u taktovima 3 i 4 nezavieno izvršavaju po 3 različite instrukcije

Slika 5



***Superpipeline*: Uvođenjem internih registara kod superpipeline računara otvoren je uslov za češće očitavanje instrukcije i smenjivanje mašinskog ciklusa, EX je izvršna faza.**

Slika 6



VLIW *pipeline*. Svaka instrukcijske reč kod VLIW računara nosi dovoljno informecije da istovremeno zaposli više izvršnih jedinica, u ovom slučaju tri

Slika 7



Vektorski *pipeline*: Jedne vektorske instrukcije primenjuje se nad nizom različitih podataka (nad vektorom), dužina ,stepenice' izvršnih faza zavisi od veličine vektora i interne ogranizacije, primer je za vektor od šest elemenata

oblikovenje novih arhitektura. Sa druge strane, RISC koncept koji teži da minimizuje hardverske resurse odstranjivanjem svega što ne doprinosi performansama, ostavlja prostor da se na čipu nađu svi potrebni resursi za ostvarenje superskalarne arhitekture. Dalje poboljšanje tehnologije dovodi do ugrađivanja jedinica za podršku grafičkih operacija (primer i860), korišćenje pune 64-bitne arhitekture (MIPS R4000) ili raznih tipova keš (cache) memorije na čipu. Prema nezvaničnim informacijama, Intelov i586 koji je iznikao iz CISC osnove će imati superskalarnu, 64-bitnu arhitekturu i odvojene keš memorije za instrukcije i podatke, što samo po sebi dovoljno govori o pravcima koji najviše doprinose performansama, čak i kada je u pitanju procesor zasnovan na zastarelim idejama.

RISC arhitekture odlično se snalaze u novim tehnologijama. GaAs mikroprocesori se baziraju na RISC konceptima prilagođenim ograničenjima brzog materijala kakav je GaAs. Primer takvog procesora je DARPA (Defense Advanced Research Projects Agency – agencije za napredna istraživanja američkog Ministarstva odbrane) čip koji, u novoj varijanti ima specifikacije: takt 250 MHz, 170 MIPS a E/D – MESFET tehnologije, namenjen prvenstveno za UNIX sisteme. Takav brz procesor zahteva dva nivoa keša, jedan na čipu i jedan van njega, u CMOS tehnologiji. Pri pomenutim brzinama jedan od glavnih problema je fizičko rastojenje između komponenata, naročito procesora i memorije gde svaki suvišni centimetar degradira performanse, čega je bio svestan i genijalni Seymour Cray pri gradnji istoimenih računara. U GaAs tehnologiji se rešenje vidi u korišćenju MCM (Multi Chip Module) pakovanja koje više čipove stavljaju u isto kućište.

KAO SIMFONIJSKI ORKESTAR

Vratimo se multimedia problemu. Jedno od predloženih rešenja je korišćenje RISC jezgra sa većim brojem akceleratora za pojedine oblasti. Tri osnovna tipa ubrzivača bi bili akceleratori osnovnih procesa, sistemskih funkcija i aplikativnih funkcija. Za ubrzivače osnovnih funkcija mogu se, na primer, koristiti nizovi procesora koji komuniciraju preko deljenih memorija. Međutim, primena takvih sistema još uvek nailazi na otvorene probleme kakvi su problemi keširanja u multiprocesorskim sistemima sa deljenom memorijom. U prošlosti su se što zbog problema softverske, što hardverske prirode, multiprocesorski sistemi uglavnom nalazili po istraživačkim laboratorijama. Profesor Univerziteta u Mančesteru Devid Aspinell, koji je učestvoveo u pravljenju čuvenog Atlas računara pre tri decenije i bavio se razvojem paralelnih računara, izneo je mišljenje o malom komercijalnom efektu takvih sistema u obliku anegdote. Pre nekoliko godina, na jednom popularnom predavanju, uporedio je jednoprocesorski računar sa solistom, a multiprocesorski sa simfonijskim orkestrom. Posle predavanja, prišao mu je slušalac upućen u problematiku i upitao ga: „Koliko simfonijskih orkestara donosi profit?" Danas se stvari menjaju nabolje. Čak i najkonzervativnije kompanije predstavljaju modele malih računara baziranih na multiprocesorskom konceptu, koji je vrlo pogodan za zahtevne multimedijalne aplikacije.

Među pogodnim akceleratorima koji se mogu koristiti i kao dodatak za PC treba istaći hardversku podršku animaciji koja koristi algoritme kompresije slike do 1:200 i bazira se na novoj Intel DVI (Digital Video Interactive) seriji čipova i korišćenje adaptivnih sistema za promenu parametara disk keširanja čime se korišćenje disk keša podređuje tipu podataka koji se prosleđuje od/ka računara. Brz prenos podataka je jedan od osnova za multimedia koncept. Osim toga, pretraživanje i efikasno dobavljanje ogromne količine raznorodnih informacija su ključni zahtevi za buduće sisteme. U slučaju uskladištenih hiljada sati audio, video i tekstualnih materijala na CD ROM ili drugim memorijama, biće neophodno za vrlo kratko vreme pretražiti i pribaviti odgovarajući video insert po raznim osnovama pretraživanja, obezbediti prateći audio signal i tekst, obraditi ih i uklopiti u aplikaciju. Neke od ovih problema pomenuo je profesor Ali Hurson sa University of Pennsylvania. Po njegovim rečima, CRAY 1 treba za pretraživanje baze podataka US Patent Office od 25 Terabajta čak 500 časova! Zbog takvih paradoksa, on predlaže korišćenje asocijativne memorije koje su skuplje, ali svoje prednosti pokazuju u bazama podataka.

U čemu je budućnost? U umnožavanju funkcionalnih jedinica, kombinacijama sa VLIW, masivnom paralelizmu i eksploataciji RISC ideja. Za kraj, valja pomenuti da su učesnici imali priliku da se upoznaju sa strukturom i osobinama perjanice IBM-a u oblasti radnih stanica, RS/6000. Međutim, po tvrdnjama jednog od učesnika u stvaranju te mašine, osnovni koncepti inspirisani su rešenjima sa davnih CDC 6600 i IBM 360/91 računara! Dakle, za jasnije viđenje budućih računara treba dobro proučiti već podebelu istoriju računarstva, jer se možda baš tu kriju prave ideje!

C I VIŠE OD C-a

Windows-ima je danas teško odoleti. Po najnovijim podacima, prodato je 2,750,000 primeraka ovog programa i od toga je oko 1,500,000 u redovnoj upotrebi. Širenja Windows-a u ovom trenutku izgleda nezadrživo i svakako je trend devedesetih. Sada, tačno godinu dana posle predstavljanja Windows-a 3.0, postoje sasvim dobri korisnički programi iz svih uobičajenih oblasti. Izvesna praznina postojala je na tržištu profesionalnih pravodilaca, ali pristižu i profesionalna oruđa za programere...

Duško Savić

Sada imamo *Smalltalk/V for Windows*, *Turbo Pascal for Windows*. Logično je bilo da *Borland* prepravi svoju i inače dobru kombinaciju *C*-a i *C++*-a i napravi verziju koja podržava *Windows*-e. Izgleda da je naziv „Borland" u imenu od sada rezervisan za profesionalne verzije paketa. Tako smo umesto *Turbo C++ Development Edition* u verziji 1.0 dobili *Borland C++* verziju 2.0. *Turbo C++* se i dalje prodaje nezavisno od *Borland C++*.

Borland C++ dolazi na sedam disketa prečnika 5.25 inča i kapaciteta 1.2 megabajta, odnosno, na jedanaest disketa prečnika 3.5 inča i kapaciteta 700 kilobajta. Na obe grupe disketa je isti program. Priručnici su teški 5.5 kilograma i jedna grupa se odnosi na *Borland C* i *C++*, a druga je za pomoćne programe *Turbo Assembler*, *Turbo Debugger* i *Turbo Profiler*. Osnovni priručnik je i najmanji. *Getting Started* na samo 150 strana objašnjava čemu služi paket, koji su mu sastavni delovi i predstavlja i uvod u *C++*. *Programmer's Guide* (444 strane) je već ozbiljnija literatura: definiše jezike *C* i *C++*, objašnjava povezivanje sa asemblerom i upotrebu novih sistemskih klasa za različite vrste objekata. *User's Guide* (308 strana) bavi se editorom, stvaranjem *Windows* aplikacija, pomoćnim programima i objašnjava šta su prevedena zaglavlja (precompiled headers). Najdeblja knjiga, sa svoje 592 strane, je *Library Reference* i sadrži detaljno objašnjenje za skoro 500 funkcija koje *Borland C++* nudi. Preostale dve knjige su čudesno tanke: 88 strana za *Help Compiler* i 130 strana za *Whitewater Resource Toolkit*.

Spisak literature ovim nije iscrpljen. *Turbo Debugger* (426 strana) je sada u verziji 2.5, baš kao i *Turbo Assembler* (494 strane). Najmanji „profesionalni" pomoćni program je *Turbo Profiler*, u verziji 1.1 (samo 194 strane).

Važno je shvatiti da *Borland C++* može da pravi i DOS i *Windows* aplikacije, ali da se uvek izvršava kao DOS aplikacija. Zato se instalacija pokreće iz DOS-a, uobičajenim A:INSTALL. Za instalaciju moramo upotrebiti program INSTALL jer su datoteke na disketama arhivirane. INSTALL se svodi na niz naredbi UNZIP koje istovremeno dearhiviraju i kopiraju na tvrdi disk.

Za testiranje je poslužio AT sa 12MHz bez stanja čekanja, sa dva megabajta centralne memorije, i sa samo dva megabajta prostora na tvrdom disku. Instalacija svih datoteka zauzela je čak 15 megabajta prostora na tvrdom disku(!)



U zreloj fazi: Asembler, profajler i dibager

MEŠANI JEZICI

Borland C++ je istovremeno i ANSI C i „obični" C i C++. ANSI je svetski standard za C, „obični" C je *Borlandove* dopuna tog standarda a C++ je nadole kompatibilan sa ANSI C-om i, naravno, predstavlja svet za sebe. Na nivou C-a, omogućan je prelazak sa *Microsoft C*-a, što u normalnim okolnostima nije naročito bitno, ali se ne može izbeći u proizvodu sa *Windows* usmeranjem.

Situacija je, naime, sledeća. Majkrosoftov C je osnovna alatka za pisanje *Windows* programa ali nije dovoljan već se mora dokupiti i niz pomoćnih programa i biblioteke, pod zajedničkim imenom *SDK for Windows* (*System Development Kit*). U komercijalnom smislu za Majkrosoft je takvo razdvajanje funkcija više nego povoljno: prodato je 50.000 primeraka paketa *SDK*!

Da bi se napisao program pod *Windows*-ima, treba se upoznati sa tzv. *Windows API* funkcijama – a njih ima oko 600. Postoje tačna pravila po kojima ih treba pozivati, tako da je 1) programiranje za *Windows*-e teško samo po sebi, i 2) zahteva da se direktno od Majkrosofta kupe dva odvojena proizvoda. Takva rešenja ne dopadaju se milionima programera koji ne konste *Microsoft C* ili uopšte ne konste C, a o osećaju zavisnosti od jednog jedinog proizvođača da se i ne govori. Zato je ideal svih drugih softverskih kompanija postalo objaviti programski jezik koji ne zahteva ni *Microsoft C* ni *SDK for Windows*. Koliko proizvođača – toliko rešenja. Tako na primer, *JPI Modula-2* u verziji 2.0 može da stvori *Windows* program ali se prisustvo programa *LINK* iz *SDK* podrazumeva, što je slučaj da *Microsoft C* nije potreban ali *SDK* jeste. Sa svoje strane, čisto objektni jezici *Actor* i *Smalltalk/V* ne zahtevaju ni *Microsoft C* ni odgovarajući *SDK*.

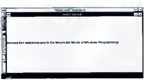
Naravno, onih 600 *API* funkcija se moraju na ovaj ili onaj način naći u svakom paketu, pa ako želimo da pređemo na neki od spomenutih jezika, moramo biti svesni da jedno učenje zamenjujemo drugim. Umesto 600 funkcije učićemo potpuno nove jezike, što može ali i ne mora biti prihvatljivo za prosečnog programera. Da stvar bude još lepša, pobrinuo se sam *Borland*, nedavnim objavljivanjem *Turbo Pascala for Windows* (*TPW*). *TPW* je konglomerat prethodno navedenih koncepcija. Ako želimo samo program koji radi kao prozor ali bez specijalnih efekata koji *Windows*-e čine zanimljivim, *TPW* je najbolji izbor. Ako želimo da iskoristimo *Windows*-e do kraja, imamo dve ravnopravne mogućnosti: da se sazivimo sa spomenutih 600 funkcija ali „prepevanih" na paskal ili da naučimo potpuno novi jezik koji je u obliku hijerarhije nazvane *ObjectWindow* isporučen sa *TPW*-om. Slična rešenja – nove hijerarhije objekata specijalno za pisanje *Windows* programa – nalaze se i uz neke komercijalna proizvode (*Zortech C++*) ili se kupuju posebno, za svaki od popularnih *C++* prevodilaca.

Bilo bi takođe logično očekivati da se prevodilac koji pravi *Windows* program izvršava i sam kao *Windows* program, jer se time produktivnost podiže na viši nivo. To je zaista slučaj za *Zortech C++*, *Actor*, *Smalltalk/V* i *Turbo Pascal for Windows*. Razni jezici firme *JPI* (*Modula-2, C, Pascal, C++*), *Borland C++* i sam *Microsoft C* izvršavaju se iz DOS-a a mogu proizvoditi DOS ili *Windows* programe. Pri tome je bitno shvatiti da se programi za DOS ne mogu prostim naredbama programskom jeziku pretvoriti u program za *Windows*-e. Nije moguće „ubiti dve muve" istim programskim paketom! To se pogotovu odnosi na *Borland C++*, jer on zaista sadrži hijerarhije objekata, ali ne za *Windows*-e. Čak se otvoreno preporučuje kupovina literature za *SDK for Windows* pa je samo delimično tačna *Borlandova* reklama da se ovim paketom mogu pisati *Windows* aplikacije a da se na konsti SDK. No, ta reklamna poruka nije ni sasvim netačna, jer je dovoljno uzeti bilo koju knjigu o programiranju za *Windows*-e umesto literature za SDK.



Snimio: Nenad Marijanović

Rame uz rame sa Microsoftom: *Borland C++*



Rezultat programa WHELLO, dugačkog preko 200 linija



Osnovni izgled Resource Managera

Dakle, u skoroj budućnosti, ali ne sada i odmah, i sa ovim paketom pisaćemo *Windows* programe. Pogledajmo ipak kako izgleda rad sa ovim što danas *Borland* nudi.

PROZORI...

Sledeći uputstvo iz priručnika, program od kojeg početnik za *Windows*-e treba da počne je WHELLO, koji otvara prozor pod *Windows*-ima i ispisuje jednu rečenicu u njemu. Tako jednostavnom zahtevu trebalo bi da odgovara jednostavan program, ali to nije slučaj. WHELLO ima preko dvesta redova i sastoji se isključivo iz poziva *Windows* funkcija. Sve se to prevodilo 4-5 minuta uz stalno angažovanje diska. Manji deo prevođenja otišao je na sam C, a veći na rad sa RC prevodiocem. Možda i *Microsoft C* tako pravi *Windows* programe, ali sve je to mučenje u odnosu na prave *Windows* jezike, kao što su *Turbo Pascal for Windows*, *Actor* i *Smalltalk/V Windows*.

Borland C++ generiše i DLL datoteke, a to su biblioteke funkcija do kojih modul iz *Windows*-a ima pristup. Razlika između običnih biblioteka i DLL-a je u trenutku povezivanja. API funkcije iz DLL-a povezuju se sa ostatkom programa za vreme izvršenja programa (dinamičko povezivanje), umesto za vreme prevođenja (statičko povezivanje). DLL datoteke u principu imaju smisla i nezavisno od *Windows*-a, i postoji bar jedan jezik koji ih kao takve i pravi, direktno pod DOS-om (*JPI Modula-2 2.0*).

Windows programi se uglavnom sastoje iz DLL datoteka. Drugi značajan element su resursi.

... I RESURSI

Resurs je neki grafički element koji se u programu koristi ali nije sastavni deo programa. Postoje različite kategorije resursa: tačkaste (bit-by-bit) slike, oblik kursora, pravougaonici za dijaloge, ikone, meniji i tekst itd. Isti resurs može biti upotrebljen u potpuno različitim programima i obratno: izmenom resursa menjamo izgled programa bez ikakvog programiranja, pa čak i ako nemamo tekst originalnog programa! Resurs sam za sebe ne definiše šta program radi, ali utiče na ono što korisnik vidi i može da uradi.

Pošto su resursi nezavisni od programa, u Majkrosoftov SDK je uključen RC prevodilac, tj. poseban prevodilac koji za svoje ulazne podatke uzima ASCII datoteke sa opisom resursa i prevodi ih u oblik koji *Windows* programi razumeju. *Borland* je licencirao Majkrosoftov *Resource Compiler* iz *SDK*, jer se samo pomoću njega može dobiti *Windows* program iz DOS-a. RC prevodi skript datoteke (naredbe) i grafička podatke u binarni oblik, povezuje ih sa prethodno prevedenom aplikacijom, i označava da je gotova **EXE** ili DLL datoteka pripravna za rad pod verzijom 3.0 *Windows*-a.



Tipičan prizor iz cursor editora

RC se može aktivirati na nekoliko načina: iz editora opcijom **Project**, sa komandne linije DOS-a kao opcija za komandni prevodilac, ili iz make datoteke.

Borland C++ sledi *Microsoft C* i njegove načine da se iz DOS-a dođe do *Windows* programa. Zato je rad sa RC prevodiocem naporan i svodi se na uobičajeni ciklus tekstualne izmene resursa, prevođenje, snimanje, proveravanje, pa opet u krug... Lako je razumeti zašto je *Borland* priključio program *Whitewater Resource Toolkit* (WRT), jer on je potpuna suprotnost: umesto sa prevodiocem radimo direktno sa podacima koji nas interesuju. Ako je reč o ikoni, ne opisujemo je već je direktno mišem crtamo po ekranu. Umesto sa prevodiocem, radimo sa interpreterom, pa po ljubaznosti i lakoći rada WRT daleko nadmašuje Majkrosoftovu ponudu.

WRT se startuje iz *Program Managera*, i može istovremeno da radi sa dve datoteke resursa. To je zgodno zbog eventualne razmene između raznih resursa. Na vrhu ekrana vidi se glavni meni, koji istovremeno pokazuje i šta WRT može da menja: makro naredbe (**accelerator**), tačkasto definisane slike (**bitmap**), kursor (**Cursor**), dijaloge (**Dialog**), ikone (**Icon**), menije (**Menu**) i tekst (**String**). Nad ovim resursima moguća je izmena, dok WRT može učitavati i snimati ali ne i menjati tri dodatna resursa: fontove, RC datoteka i korisničke resurse.

Osnovu grafičkih resursa čini editor za crtanje, a njega u pravom objektnom maniru nasleđuju i obogaćuju editori za kursor i ikone. *Paint editor* je tipičan program svoje namene: crtanje olovkom, brisanje, „razmazivanje", inverzija, veliki izbor boja i sl., a sve se to dešava nad znatno manjom slikom nego npr. u *Windows Paint*-u.

Nema potrebe detaljno opisivati ostale resurse, jer ih možemo menjati kompletno i lako.

UPOMOĆ!

Help prozor je jedno od elegantnih i korisnih sredstava koja olakšavaju život sa *Windows*-ima. U svakom trenutku, u svakoj profesionalnoj aplikaciji možemo – najčešće pritiskom na taster **F1** – pozvati *Help* prozor na ekran i iz njega pročitati ili pronaći odgovor na neko pitanje. Dodatni kvalitet *Borland C++*-a je mogućnost uključivanja istog takvog *Help* prozora i u programe koje sami budemo pisali. *Help* prozor ima dva aspekta: generisanje teksta i saradnja sa aplikacijom iz koje se poziva. Za prvo je zadužen tzv. autor *Help* sistema, i njegova odgovornost se sastoji u planiranju, pisanju i kodiranju tema. Ovo drugo je briga programera *Help* sistema: on programira direktno na C-u. Priručnik detaljno opisuje šta treba uneti u program da bi se posle pritiska na **F1** ili **Shift-F1** pojavio *Help* prozor. Inače, svaki Help prozor je aplikacija potpuno nezavisna od bilo koje druge pod *Windows*-ima.

Autor *Help* sistema raspolaže osnovnim naredbama koje unosi u tekst. Programska podrška *Help* prozora je zapravo interpreter tih naredbi. Osim toga, autor mora dobro znati šta hoće da postigne svojim porukama, tj. mora znati kome se obraća, koji je nivo korisnika aplikacije itd.

EDITOR

Editor je sasvim sličan onom iz *Turbo C*-a, verzija 1.0: višestruki prozori, miš, datoteka praktično neograničene dužine, kretanje kursora po *Wordstar* standardu, izmene teksta po ugledu na *Windows* editore. Opcije glavnog menija su: *sistem* (obaveštenja o editoru, brisanje svih prozora), **File** (datoteke i imenici, odlazak u DOS, štampanje), **Edit** (blokovi teksta, prenos u pomoćni prozor *Clipboard*, povratak obrisanog), **Search** (izmene i pretraživanje, lociranje grešaka, prelaza na red sa zadatim brojem), **Run** (izvršenje programa odjednom, korak po korak, u koracima ali sa preskakanjem funkcija), **Compile** (prevođenje), **Debug** (razgledanje aktuelnih podataka po strukturama, postavljanje zaustavnih tačaka), **Project** (otvaranje i zatvaranje projekta, dodavanje elemenata u projekat), **Options** (postavljanje opcija za prevođenje, povezivanje, vrsta generisanog programa itd.), **Window** (prozori) i **Help** (pomoć).

Sva to liči na druge *Borlandove* editore novijeg datuma, ali ima i novih mogućnosti. Osim naredbe *undo* sada postoji i *redo*, što znači da ne samo da možemo poništavati efekat prethodne radnje nad tekstom, nego da možemo poništiti i poništavanje. Značajno je i izdvajanje nova opcija **Project**. Ona ima smisla za programe koji se sastoje iz velikog broja datoteka. Naglašavanje takve opcije postalo je nužno jer se – hteli mi to ili ne – programi za *Windows*-e sastoje iz mnoštva takvih datoteka, pri čemu su npr. resursi ravnopravni delovi programa, a opcija **Project** može pokrenuti ne samo prevodilac za *C++* nego i RC prevodilac i sve druge potrebne programe.

Dakle, nove opcije i mogućnosti u editoru – ali to nije sve! *Borland C++* sadrži čak četiri verzije prevodioca: dve editorske i dve koje se startuju iz DOS-a. Razlike su u korišćenju memorije. Program BC je „običan" editorski prevodilac, tj. prvo se učitava editor sa tekstom programa, a onda prevodimo i testiramo direktno iz njega. BC zna za samo 640K centralne memorije, pa je veličina programa koje iz njega možemo razvijati prilično ograničena. Pandan ovom programu je BCX, a to je opet editorski prevodilac s tom razlikom što radne tabele pamti u produženoj (extended), proširenoj (expanded) memoriji ili RAM disku (ako ovaj postoji). Razlika između ova dva prevodioca je drastična. BC je „svestan da je mali", i automatski tokom prevođenja snima neke delove programa na disk. Ako na disku nema mesta, prevođenje se prekida.

Evo kako to izgleda na primerima koji se nalaze u samom paketu. Program TCALC je dinamička tabela, sa ukupno 10592 programska reda. BC je to preveo za petnaestak minuta, a kod ponovljenog prevođenja radio je sedam minuta dok mu nije ponestalo prostora na disku, dotle je preveo 8421 red. Nasuprot tome, BCX je ne samo uspeo da sve prevede, nego je to uradio za samo 70 sekundi! I taj uspeh ima svoju cenu: bilo je neophodno osloboditi produženu memoriju od cache programa. No zato je prevodilac posle tih 70 sekundi prijavio slobodnih 1167K, za razliku od programa BC, u kojem se količina slobodne memorije često približava nuli, a tada nastaje prenos na disk uz svo usporenje koje iz toga proizilazi.

Opšte je poznato da se program u C-u piše kao niz datoteka: jedna je zaglavlje (**header**) i sadrži deklaracije, a druga je prirodni nastavak prve i sadrži izvršne naredbe, tj. funkcije. Isto zaglavlje se može upotrebiti u različitim funkcijama, ali običan C prevodilac će svaki put iznova zaglavlje učitati sa diska i prevesti ga. Ako deset raznih funkcija koristi isto zaglavlje, proces se ponavlja deset puta a vreme prevođenja se produžava za devet. *Borland C++* zato uvodi prevedena zaglavlja (**precompiled headers**): zaglavlje se prevodi samo prvi put, zatim se njegov prevedeni oblik stavlja u posebnu datoteku na disku. Sva prevedena zaglavlja nalaze se u istoj – potencijalno velikoj – datoteci. To ubrzava prevođenje istog programa četiri do pet puta.

KLASE I HIJERARHIJE

C++ je, naravno, objektni jezik, pa bi se moglo očekivati da se uz sam prevodilac nađe i nekoliko objekata. I dok glavnog zgoditka – hijerarhije klasa za *Windows*-e – još uvek nema, ne znači da *Borland* ovog puta nije ponudio ništa. Isporučena hijerarhija objekata bavi se često korišćenim, tzv. klasičnim strukturama podataka, kao što su liste, stakovi, redovi i dr. Osnova hijerarhije za većinu klasa je objekat **Object**, ali postoje i druge, od ovog objekta nezavisne klase, npr. **ContainerIterator**, **DoubleListElement** i **ListElement**. Te nezavisne klasa su samo ispomoć za osnovnu hijerahiju.

Hijerahija klasa izgleda kao u tabeli 1.

Ako ova poslednje tri klase odbacimo kao pomoćne, hijerarhija za **Object** sinktno se deli na klase koje ne sadrže i na klase koje sadrže druge objekte. Prvoj grupi pripadaju klase **Error** (greške), **Sortable** (ono što se može sortirati), **String** (tekst), **Association** („labava" veza elemenata) itd. Podklasa **Container** prirodno se deli na **Collection** (kolekcija elemenata) i tri vrste redosleda: **Stack**, **Queue** i **Deque**. Redosledi imaju sledeća zajednička svojstva: 1) objekti se mogu umetnuti ili obrisati 2) redosled umetanja ili brisanja je bitan 3) umetanje ili brisanje dešava se na tačno određenom mestu. Dodatno, postoji funkcija **isEmpty**, kojom doznajemo da li je niz prazan ili ne.

Ovako opisani nizovi imaju zajednička svojstva, ali ona nisu dovoljno jaka da bi **Stack**, **Queue** i **Deque** imeli zajedničku nadklasu. Suprotno tome, najčešće korišćene klase su sadržane u klasi **Collection**. To pruža značajnu fleksibilnost: sve dok tokom rada sa programom ne saznamo tačno koja nam kolekcija treba, možemo koristiti opštiju klasu **Collection** bez ikakvog gubitka funkcionalnosti.

Kolekcije se prirodno dele na uređene i neuređene. Neuređene kolekcije su **HashTable** i izvedene klase **Bag**, **Set** i **Dictionary**, kao i **List** i **DoubleList**. Bilo koji objekat iz hijerarhije za **Object** može biti član neuređene kolekcije. U uređenim kolekcijama poredak elemenata je bitan pa su kao članovi ovih objekata dozvoljeni samo elementi klase **Sortable**. U uređena kolekcije spadaju **AbstractArray**, i njegove podklase **Array** i **SortedArray**, tj. klase koje su ekvivalentne običnim nizovima.

Za svaku kolekciju mora postojati iterator, tj. naredba kojom se „prolazi" od prvog do poslednjeg člana strukture. Rešenje u *Borland C++*-u na prvi pogled nije logično, jer su iteratori u drugim objektnim jezicima takođe podklase u odnosu na **Object**. Međutim, sve postaje lako ako se prisetimo da u *C++*-u postoji pojam „prijatelja", tj. da se može dozvoliti nekoj funkciji koja nije iz klase da se obrati internim podacima te klase. Konkretno, apstraktna klasa **ContainerIterator** je prijatelj (**friend**) apstraktne klase **Object** i zato je moguće da iteratori klasa iz **Object** hijerarhije ne budu i sami njene podklase.

Tabela 1

- **Object**
  - **Error**
    - **Sortable**
      - **String**
      - **BaseDate**
        - **Date**
      - **BaseTime**
        - **Time**
    - **Association**
    - **Container**
      - **Collection**
        - **AbstractArray**
          - **Array**
          - **SortedArray**
        - **HashTable**
          - **Bag**
            - **Set**
              - **Dictionary**
        - **List**
        - **DoubleList**
      - **Stack**
      - **Queue**
      - **Deque**
  - **ContainerIterator**
    - **HashTableIterator**
    - **ListIterator**
    - **DoubleListIterator**
    - **ArrayIterator**
  - **DoubleListElement**
  - **ListElement**

Hijerarhija klasa: uvlačenje redova znači nasleđivanje

Korisna adresa

MARAND d.o.o.
Kardeljeva ploščad 24, 61000 Ljubljana
Tel. (061) 340-652, 371-114
Fax (061) 342-757

USAVRŠENI TRIO

Preostala tri programa: *Turbo Profiler*, *Turbo Debugger* i *Turbo Assembler* su dopune već dobrih i zrelih programa. *Profiler* meri gde program troši vreme, *Debugger* pomaže u otkrivanju grešaka i nepravilnosti u radu programa, a *Assembler* je sredstvo za pisanje mašinskih programa. O njima nećemo mnogo ovom prilikom, jer je ovaj časopis više puta opisivao ove programe. Spomenimo samo novine: *Profiler* sada može da čita i prikazuje tekstualne datoteke. Glavna novost u *Debuggeru* je mogućnost rada sa Windows programima. Nažalost, *Debugger* nije poseban prozor pod *Windows*-ima (to bi bilo dragoceno tokom razmatranja *Windows* aplikacija uživo). Za uzvrat, moguć je tzv. *remote debugging*, tj. izvršavanje programa na jednoj mašini a rad dibagera na drugoj, povezanoj serijskim kablom. Parametri te veze mogu se direktno menjati. *Debugger* takođe prikazuje informaciju o statusu *Windows* aplikacije: koje poruke prima prozor, lokalnu i dinamičku (heap) memoriju, spisak modula iz kojih se sastoji aplikacija.

Sve ove novosti su dobrodošle, ali konkurencija to radi bolje. U jezicima *Smalltalk/V for Windows* i *Actor* debugger je još samo jedan prozor, ravnopravan sa ostalima, i sa njime se radi interaktivno. Naravno, možemo staviti primedbu da su ova dva jezika ipak interpreteri (iako oba omogućavaju pravljenje zatvorenih aplikacija) i da im je zato lakše. Primedbe takve vrste ne zanimaju mnogo krajnjeg korisnika, tj. programera, koji želi brz pristup svojim podacima i to naročito u kriznim trenucima.

Nova u *Turbo Assembleru* kompletna podrška i za procesor 486. Takođe je podržan *DOS Protected Mode Interface* (DPMI), što znači da ovaj program može da radi u zaštićenom režimu (protected mode) pod *Windows*-ima. Zato postoje dve verzije asemblera: TASM i TASMX. TASMX može da se izvršava pod *Windows*-ima na procesoru 386 u proširenom (enhanced) režimu. To ima smisla ako je na raspolaganju mnogo produžene memorije, i ako treba prevoditi vrlo velike module. Za sve ostale upotrebe, TASM je mnogo brži, a takođe se može izvršavati pod 386 *Windows*-ima.

Kao i pre, i u *Borland C++* se može direktno postaviti asemblerski kod: mešanje je dozvoljeno posle naredbe asm. Isto rešenje primenjeno je i u novijim verzijama *Turbo Pascala*.

Ovim paketom je *Borland* stao uz bok Majkrosoftu. *Windows*-a što se tiče, *Borland C++* ekvivalentan je svom uzoru, *Microsoft C*-u zajedno sa *SDK*-om. No, bliskost originalu je isuviše velika i od elegantnog rada sa *Windows*-ima nema ništa! Stoga je najavljena hijerarhija *ObjectWindows*, koja bi trebalo da bude funkcionalno identična onoj u *Turbo Pascal for Windows*. U međuvremenu, na *Borlandovoj* strani su i cena (dvostruko manja od cene za *Microsoft C + SDK*) kao i činjenica da je ponuđen *C++* sa nekoliko korisnih objekata, dok je *Microsoft C* i dalje samo C. I, naravno, bez obzira na trenutne probleme sa *Windows*-ima, *Borland C++* je bio i ostao odlično sredstvo za razvoj programa pod DOS-om.

KRALJ JE STIGAO

Dugo najavljivan i još duže očekivan *Clipper 5.0* je nedavno stigao i kod nas. Pre nego što su korisnici imali vremena da kažu „A", izašao je *Clipper 5.01*. Njegova pojava će verovatno rasčistiti dilemu: preći na novu verziju ili ostati na staroj i isprobanoj *Summer '87*.

Nenad Batoćanin

Clipper 5.0 je doneo zaista velike i vredne novine – radi se o kompletnom, izuzetno zamišljenom sistemu za razvoj baza podataka. Ograničenja koje DOS nameće elegantno se prevazilaze a moćan jezik omogućava veoma produktivno programiranje. Međutim, veliki sistem je sa sobom doneo i ono što ga neminovno prati: greške. Tako je verzija 5.0 zapamćena kao nesigurna za korišćenje. Kada se nedavno pojavila verzija 5.01, očekivalo se da će se u njoj ispraviti greške verzije 5.0. Ali, *Clipper 5.01* je otišao mnogo dalje – veliki broj novih i ponovo napisanih funkcija, poboljšani svi kritični sistemi i potpuno novi *Debugger*.

U velikoj armiji korisnika programa koji su napisani na *Clipper*-u, najviše je osoba bez predznanja iz programiranja. Iz samo tog razloga sve *Clipper* aplikacije bivaju uvek testirane u najstrožijim uslovima – ono što programu mogu uraditi neiskusni korisnici, teško mogu smisliti i iskusni sa najgorim namerama! Zato su *Clipper* programeri veoma uzdržani prema novoj verziji *Clipper*-a: stara verzija dobro radi, ima grešaka, ali su se navikli na njih i naučili su da ih zaobilaze. A nove stvari uvek donose i nove greške, a time i nove probleme. Ne treba zanemariti i prepravke starih programa koji rade, pri čemu može nastati gomila novih komplikacija. Ipak, svet se okreće – treba krenuti dalje od '87!

NOVI JEZIK

Clipper je davno prestao da postoji kao običan *dBASE* prevodilac i posleo samostalni proizvod, tako da su inženjeri *Nantucket*-a imali potpunu slobodu pri izmeni sintakse jezika. Međutim, mudro je bilo ne promeniti *Clipper* potpuno – bolje je napraviti jezik kompatibilan sa prethodnim verzijama i tako zadržati pravu armiju starih korisnika. Tako su i postupili: novi *Clipper* u potpunosti podržava sve naredbe iz verzije *Summer '87*, tako da bi svi vaši programi morali da rade bez ikakvih dodatnih izmena i kad ih prevedete novom verzijom. Znači, što se tiče sintakse, možete preći na novu verziju i koristiti je i bez učenja novih naredbi.

Letimičan pregled spiska komandi i funkcija daje utisak da se ništa naročito nije promenilo – sviega nekoliko novih komandi i funkcija. Međutim, izgled i te kako vara: novi *Clipper* ima potpuno novo "srce", koje može učiniti vaše produktivno programiranje daleko produktivnijim. Na slici 1 prikazan je primer programa pisanog na novom *Clipper*-u. Vidi se niz novina: pretprocesorske direktive, novi načini komentarisanja, kodni blokovi, *in-line* dodeljivanje, itd.

Naizgled benaina stvar kao što je naredba za komentarisanje može biti vrlo važna za pregledan program. Novi *Clipper* podržava čak pet načina komentarisanja programa: na početku reda komentar može početi sa "*" ili sa "NOTE", bilo gde u redu sa "&&" ili sa "//", a blok se može komentarisati korišćenjem "/*" i "*/". Uvođenje mogućnosti komentarisanja bloka i korišćenjem "//" umesto "&&" značajno se olakšava komentarisanje programa. Ovu jednostavnu novinu nikako ne treba potceniti – videćete i sami koliko je ugodnije kucati "//" umesto "&&" i komentarisati blok samo dodavanjem oznake na početak i kraj bloka.

Dosadašnjim naredbama dodeljivanja **STORE** i "=" dodata je i nova naredba ":=" za *in-line* dodeljivanje, slična odgovarajućoj naredbi u C-u. Dodeljivanje se može vršiti i u okviru neke druge naredbe, što sigurno neće doprineti preglednosti programa, ali će rezultovati kraćim kodom. Na primer

```
? X := (Y := 10) + 2
```

U ovoj naredbi promenljiva **Y** prvo dobije vrednost 10, zatim se promenljivoj **X** dodeljuje vrednost **Y+2=12**, a na kraju će se odštampati vrednost promenljive **X**. Pomoću *in line* dodeljivanja mogu se dodeljivati vrednosti i celih nizova, na primer

```
Array := {1, 2, 3, 4}
```

tako da sada nije potrebno pisati zamorne redove oblika **ARRAY[I]=**... Što se tiče označavanja nizova, novi *Clipper* će čak prihvatiti i sledeću konstrukciju

```
{11, 22, 33, 44}[3]
```

koja označava treći element navedenog niza (33). Kao značajna novina, prvi put su uvedeni i višedimenzionalni nizovi. Na primer, naredba

```
PRIVATE X[10,5,5]
```

kreira trodimenzionalni niz **X** sa dimenzijama 10×5×5. Nizovi se mogu kreirati dinamički, tako da dimenzije ne morate fiksirati za vreme pisanja programa. Referisanje na elemente niza se obavlja na uobičajen način

```
Y := X[2,3,4]
```

I višedimenzionalni nizovi se mogu specificirati korišćenjem prethodne notacije. Na primer, naredba

```
X := {1, {21, 22, 23}, {31, 32, 33}}
```

pretvara **X** u niz sa tri elementa: prvi je broj 1, a drugi i treći su nizovi od po tri elementa. Naredba **? X[2,3]** ispisuje vrednost 22.

Procedure i funkcije sada mogu da vraćaju kao rezultat i cele nizove. Na primer, možete pisati

```
FUNCTION FArray (ArrayNo)
    x := {1, 2, 3}
    y := {4, 5, 6}
RETURN IF (ArrayNo==1, x, y)
```

Navedena funkcija **FArray** u zavisnosti od parametra vraća niz **X** ili **Y**. Ovu funkciju možemo ravnopravno koristiti u izrazima. Na primer, ako napišemo naredbu **? FArray(1)[3]** od štampaće se treća vrednost niza **X**.

Treba imati u vidu da su operacije sa nizovima izvedene preko pokazivača. Tako je ono što mi vidimo kao niz u stvari pokazivač na početak niza. To implicira mogućnosti koje u sebi kriju i opasnosti. Na primer

```
x := {1, 2, 3}
y := x
y[1] := 11
? x[1]
```

Navedeni program štampa kao rezultat broj 11 a ne 1, kako bi se očekivalo. To je zato što naredba **Y := X** u suštini samo iskopirala pokazivač (adresu), a ne ceo niz. Nizovi se fizički kopiraju korišćenjem funkcija **AClone** i **ACopy**.

Da bi vaši programi izgledali jednoobrazno i da bi se razlikovale naredbe za poređenje promenljivih i dodeljivanje vrednosti, preporučljivo je da potpuno prestanete sa upotrebom starih naredbi za dodeljivanje "=" i **STORE** i da svuda koristite naredbu ":=". Na primer, naredba **STORE 0 TO X, Y, Z** može se pisati kao **X := Y := Z := 0**. U stvari, pretprocesor u novom *Clipper*-u svaku **STORE** naredbu i zamenjuje u odgovarajući niz *in-line* dodeljivanja, tako da se može reći da naredbe **STORE** više i ne postoji.

Znalci iz **C**-a, operatori ++ i -- od sada postoje i u *Clipper*-u. Iz C-a su pozajmljene i konstrukcije /=, +=, itd. za skraćivanje naredbi popul **X:=X+1**. Kao što znate, *Clipper* je napisan na C-u. Nova verzija je očigledno po sintaksi prišla korak bliže C-u, što će se programerima sigurno svideti, bez obzira što jezik za kreiranje baza podataka po prirodi nije blizak jezicima za sistemsko programiranje. Međutim, ne treba zaboraviti da dosta korisnika *Clipper*-a nisu profesionalni programeri i da im približavanje C-u verovatno ne odgovara. Možda će presudno biti to što se nove naredbe ne moraju ni koristiti – po želji možete pisati programe potpuno isto kao i u staroj verziji.

Jedna od novina je i mogućnost pisanja više naredbi u jednom redu. Tako se naredbe

```
IF X > 4
    X += 1
ELSE
    Y /= 2
END IF
```

sad može napisati i u jednom redu

```
IF X > 4 ; X += 1 ; ELSE ; Y /=2 ; END IF
```

Simbol ";" i dalje ostaje oznaka za nastavljanje naredbe u sledećem redu – ako prevodilac nađe nešto posle oznake ";" shvatiće to kao sledeću naredbu u istom redu, a u suprotnom će potražiti nastavak naredbe u sledećem redu.

KODNI BLOK

Jedna od velikih novosti je i uvođenje takozvanih kodnih blokova (*Code Block*). To je novi tip podataka, koji predstavlja neimenovanu funkciju sa proizvoljnim brojem argumenata ▶

Listing 1

```
/* Demonstracioni program za
   nove naredbe Clipper-a 5.0 */

#include <inkey.ch>

// Nove naredbe
#command HELP <*var*> => Hlp_Lin (<*var*>)
#command REPEAT => WHILE .T.

STATIC PROCEDURE Main          // Glavni modul

LOCAL Izbor := 1, opc := {"Prva opcija", "Druga opcija", "Treca opcija"}

CBlk := {|x| x*=2 }            // Code block

REPEAT
   HELP "ovo je help linija"
   Izbor := Menu ("M1", opc, izbor++)
   DO CASE
      CASE Izbor == 0; EXIT
      CASE Izbor == 1; Func1(CBlk); LOOP
      CASE Izbor == 2; Func2(), LOOP
      CASE Izbor == 3; Func3(); EXIT
   END CASE
Nex
RETURN                         // Kraj programa
```

Novo u novom *Clipperu*: pretprocesorske direktive, novi načini komantarisanje, kodni blokovi i *in-line* dodeljivenje

Listing 2

```
PROCEDURE Main             // Glavni modul

   LOCAL Xl := 11
   PRIVATE Xp :=1

   ? Xp, Xl                // 1 11
   PP()
   PP()
   ? Xp, Xl, Xs            // 1 11 <Undef>
RETURN

PROCEDURE PP
   STATIC Xs := 0

   Xs := Xs +1
   ? Xp, Xs, "Xl"          // 1 <1,2, > <Undef>
RETURN
```

Oblasti važenje različitih klasa promenljivih

Listing 3

```
PROC Mem

PRIVATE a[4000]

FOR i := 1 TO 4000
   a[i] := Space (50000)
NEXT

QUIT
```

Ovaj program naprosto "guta" memoriju – *Clipper Summer'87* neće izdrzati više od par iteracija, dok će novi raditi sve dok imate slobodnog prostora na disku.

Listing 4

```
PROC Main
   LOCAL Ime := Space (12), Prezime := Space (15), ;
         Radi := "0", Prof := Space (10), Adresa := Space(20)
   CLS
   @ 0,0 TO 8,40 DOUBLE
   @ 1,10 SAY "Prezime stranke" GET Prezime VALID !Empty(Prezime)
   @ 2,10 SAY "Ime stranke...." GET Ime
   @ 3,10 SAY "Da li radi....." GET Radi VALID GFirma(Radi)
   @ 4,10 SAY "Zanimanje ....." GET Prof WHEN Radi $ "Dd"
   @ 5,10 SAY "Adresa stana..." GET Adresa
   READ
RETURN

FUNC GFirma (Radi)
   LOCAL GetList := {}      // Nova GET lista
   LOCAL NF := Space (20), ADR := Space(20), S
   IF Radi $ "Dd"
      S := Savescreen (3,14,6,48)
      @ 3,14 CLEAR TO 6,48
      @ 3,14 TO 6,48
      @ 4,15 SAY "Naziv firme.." GET NF
      @ 5,15 SAY "Adresa......" GET ADR
      READ
      Restscreen (3,14,6,48,S)
   END IF
RETURN .T.
```

Primer jednostavne upotrebe READ. GET naredbi

Listing 5

```
PROC Main
   LOCAL X := "Abcdefgh"          // X se ucitava
   LOCAL O := GetNew(2,4)         // Kreiranja get objekta

   CLS

   // Podesavanje parametara objekta
   O:Name := "X"                  // Ucitava se X
   O:Block := {|v| IF (PCount()>0, X:=v, X)}
   O:PostBlock := {|v| Len(v)>1} // VALID Len(x)>1

   ReadModal({O})                 // Ucitavanja
RETURN
```

Primer učitavanja jedne promenljive, ali na nižem nivou, korišćenjem operacija sa objektima

Listing 6

```
PROC Main
   LOCAL Ch
   LOCAL TB := TBrowseDB(0,0,10,65)       // Novi TBrowse objekat
   // Brisanje ekrana
   CLS
   // Otvaranja datoteke
   USE Adresar NEW

   // Dodavanja kolona TBrowse objektu
   TB:AddColumn (TBColumnNew ("PREZIME I IME", {|| Upper(Prez_Ime)}))
   TB:AddColumn (TBColumnNew ("ADRESA", {|| Adresa}))
   TB:AddColumn (TBColumnNew ("MESTO", {|| Mesto}))
   TB:AddColumn (TBColumnNew ("TELEFON", {|| Telefon}))
   TB:AddColumn (TBColumnNew ("ZANIMANJE", {|| Zanimanje}))

   // Prva kolona je zamrznuta
   TB:Freeze := 1

   // Glavna petlja
   WHILE .T.
      Ch := 0
      // Ispisivanja ekrana, prekini ako je nesto pritisnuto
      WHILE !TB:Stabilize() AND (Ch := Inkey() == 0) ; END DO

      // Ako je slika ispisana, sacekaj komandu korisnika
      IF TB:Stable ; Ch := Inkey(0) ; END IF

      // Analiza pritisnutog tastera
      DO CASE
         CASE Ch == 27 ; EXIT
         CASE Ch == 5  ; TB:Up()
         CASE Ch == 24 ; TB:Down()
         CASE Ch == 19 ; TB:Left()
         CASE Ch == 4  ; TB:Right()
      END CASE
   END DO

   CLOSE Adresar
RETURN
```

Primer reda sa *TBrowse* i *TBColumn* objektima

Kodni blok se može dodeljivati promenljivoj ili primenjivati na neki argument. To je, u stvari, funkcija nad kojom se mogu vršiti razne operacije, pošto je istovremeno i podatak. Primena ovog tipa podatka može biti izuzetna. *Clipper* ovim dobija močan dodatak standardnim makroima. Kodni blok se može dodeljivati, baš kao i svaki drugi tip podataka, promenljivoj

CB := {|X| X+5}

Ovom naredbom promenljiva **CB** dobija vrednost kodnog bloka koji specificiranu vrednost uvećava za 5. **X** je argument kodnog bloka, a **X+5** je telo koje računa vrednost. Kako se to može koristiti? Na primer, ako želimo da navedenu funkciju primenimo na neku vrednost, napisaćemo

? Eval (CB, 10)

čime se navedeni kodni blok primenjuje na vrednost 10, tako da prethodna naredba štampa 15 kao rezultat. Mogu se koristiti i funkcije **AEval** i **DBEval** koje zadati segment koda primenjuju na svaki element niza ili na svaki slog datoteke.

Osnovna snaga kodnih blokova je u mogućnosti prosleđivanja procedurama. Na primer, ako nam je potrebno više srodnih procedura, ali koje se razlikuju u nekom delu, nije potrebno da pišemo više različitih procedura. Osnovni algoritam procedure se ne mora menjati, već se promenljivi deo prosleđuje kao kodni blok. Lep primer takve primene je i nova funkcija **ASort** za sortiranje nizova, koje je postojala i ranije, ali koja sada ima kao opcioni argument kodni blok koji definiše način poređenja dva elementa niza. Time se izbegava potrebe za posebnim funkcijama/procedurama

za svako sortiranje, već se promenljivi deo specificira kao kodni blok. Na primer, sortiranje elemenata niza po YU standardu jednostavno se ostvaruje na sledeći način

```
ASort (Niz,,, {|x, y| YU(x) > YU(y)})
```

pri čemu funkcija YU na neki način vraća redosled niza simbola po YU standardu. Na taj način funkcija **ASort** "zna" kako treba da poredi elemente niza pri sortiranju: dovoljno je da primeni zadati kodni blok na elemente koje poredi. Neke od primena kodnih blokove mogu se simulirati makro-operatorom (&), pri čemu treba koristiti kodne blokova, ako je to moguće, jer se oni, za razliku od makroa, ne interpretiraju, pa su znatno brži. U praksi se može desiti da upotreba kodnih blokova (iz nepoznatih razloga) uspori program, ali je to ipak ređe. Upotreba kodnih blokova je jedan od osnovnih mehanizama u novoj strukturi *Clipper*-a.

MAKRO OPERATOR

Makro operator je i do sada bio nezamenljiv pri pisanju interaktivnih programa, pri kreiranju dinamičkih upita i u sličnim situacijama. Da podsetimo, makro operator omogućava da se proizvoljan string interpretira za vreme izvršavanja programa, čime se dozvoljava izmena dela naredbi. Jedna od stvari koje su neprekidno nervirale je ta što se makro operator mogao primeniti na promenljivu proizvoljnog sadržaja, ali ne i na izraz. To je u verziji 5.01 ispravljeno, tako da se, na primer, sledeća sekvenca

```
pom = "X > 10 .AND. " + DodatniUslov
SET FILTER TO &pom
```

sad može pisati samo kao

```
SET FILTER TO &("X > 10 .AND. " + DodatniUslov)
```

Da napomenemo, *Clipper Summer'87* nije generisao kod značajno različite dužine ako upotrebljavate ili ne makro operator, ali u verziji 5.01 postoji značajna razlika: ako ne koristite makro operatore u programu, izvršni program će biti kraći za oko 20-30K. Uopšte, novi *Clipper* odlikuje daleko modularnija struktura od prethodne verzije.

PRETPROCESOR

Možda najlepšu novinu u novom *Clipper*-u predstavlja pretprocesor. Osim standardnih direktiva **#define**, **#ifdef**, **#ifndef**, **#undef** i **#include**, on poseduje izuzetnu direktivu **#command** koja zajedno sa kodnim blokovima uvodi ogromne mogućnosti u novi *Clipper*.

Blagodeti direktiva **#include** i **#define** znaju svi C-programeri. Direktivom **#define** se definišu konstante čime se mnogo poboljšava preglednost programa. Na primer, umesto da svuda pišete **IF LastKey()=27**, napisaćete na početku programa

```
#define ESC 27
```

i onda u celom programu pišete **IF LastKey()=ESC**, što je neuporedivo preglednije. Još bolji primer je deklarisanje maksimalnih dimenzija nekog niza. Ako u celom programu pišete simboličku konstantu umesto maksimalne dimenzije, po kasnijim izmenama dimenzije moraćete da menjate samo **#define** direktivu umesto svih pojavljivanja konstante.

Direktiva **#include** služi za uključivanje datoteka u tekuću programsku datoteku. Na primer, možete kreirati neku vašu standardnu datoteku definicija konstanti i kasnije je jednostavno uključivati u program **#include** direktivom. *Clipper 5.01* poseduje dosta gotovih datoteka sa definicijama značajnih konstanti: **INKEY.CH** (kodovi tastera), **ACHOICE.CH** (konstante značajne za **ACHOICE** korisničku funkciju), **DBEDIT.CH** itd.

Direktiva **#command** je verovatno najvažniji novitet *Clipper*-a koji se tiče jezika: ona omogućava korisniku da jednostavnim i prirodnim zapisom kreira svoje komande koje će biti potpuno ravnopravne sa ostalim! Na primer, sledeće direktive dodaju **REPEAT-UNTIL** petlju osnovnom jeziku

```
#command REPEAT => WHILE .T.
#command UNTIL <lExpr> => IF <lExpr>
; EXIT ; END ; END
```

Posle ove direktive, pretprocesor svako pojavljivanje komande **REPEAT** zamenjuje tekstom "WHILE .T.", što je specificirano prvom direktivom. Druga direktive specificira da se kraj ciklusa **REPEAT-UNTIL** zameni nizom naredbi koje proveravaju uslov za kraj ciklusa. Posle ovih definicija, u programu se može pisati

```
REPEAT
    X += 2
    ? X
UNTIL X <= 100
```

potpuno ravnopravno sa ostalim komandama! Da ovo nije puka igranja, pokazuje to da su **sve** naredbe novog *Clipper*-a i realizovane preko pretprocesorskih direktiva. I najlepša stvar, sve definicije se nalaze u datoteci **STD.CH** i **potpuno su otvorene za izmene!** To znači da ne samo što možete da definišete svoje nove komande koje su ravnopravne sa ostalim, već da možete potpuno legalno da izmenite i osnovni jezik po ličnim potrebama. Ako malo pažljivije pogledate datoteku **STD.CH**, videćete da se ceo *Clipper* jezik sastoji od svega nekoliko osnovnih konstrukcija, a sve ostalo su, u stvari, pozivi funkcija maskirani pretprocesorskim direktivama. Ovakva koncepcija ima dobrih i loših osobina. Loše osobine su nemogućnost prevodioca da prijavi detaljnije opise sintaksnih grešaka u pisanju definisanih naredbi, "sasecanje" optimizatora, koji nikako ne može da obrađuje uzastopne pozive nekih funkcija i sporije prevođenje. Nesumnjivo dobre strane su mogućnost dodavanja novih komandi i jednostavna izmena osnovnog jezika. Dodavanjem novih komandi može se značajno povećati preglednost programa. Na primer, ako koristimo sledeću funkciju za realizovanje menija:

```
Izbor := Menu (opcije, broj_pocetne_opcije, invertovan)
```

možemo sledećom direktivom

```
#command CREATE MENU <options> ;
              [INITIAL <init>]     ;
              [<invert:INVERT>]    ;
              TO <select>          ;
=> ;
<select> := Menu(<options>,
```

kreirati novu naredbu **CREATE MENU IF(<.init.>,<init>,1),<.invert.>)** za ispisivanje menija. Ova nova naredba se može koristiti na bilo koji od navedenih načina

```
CREATE MENU Opcije INITIAL 1 TO Izbor
CREATE MENU Opcije INITIAL NoPoc INVERT TO Izbor
CREATE MENU {"Prva", "Druga", "Treca"} TO Izbor
```

Kao što vidite, definisana komanda je potpuno uklopljena i ne razlikuje se od ostalih naredbi, a njeno korišćenje dosta doprinosi lakšem praćenju programa.

Pri pisanju komandi složenije strukture ponekad je korisno pogledati rezultat rada pretprocesora, što se postiže opcijom /P pri prevođenju. Tako se generiše **PPO** datoteka koja sadrži izlaz pretprocesora.

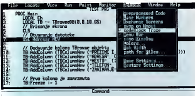
Primenjena koncepcija omogućuje da se nove verzije *Clipper*-a na nivou sintakse jezika realizuju samo kao datoteke pretprocesorskih direktiva, potpuno nezavisno od "pravog" prevoda koji *Clipper* generiše.

TIPOVI PROMENLJIVIH

Glavna boljka *Clipper*-a je sporo izvršavanje, a nastalo je kao posledica toga što se program delimično interpretira. Veliku zaslugu u sporosti imaju i dinamičke operacije sa promenljivim: kreiranje i svi pristupi promenljivim su se u verziji *Summer '87* obavljale za vreme izvršavanja programa, a to je daleko sporije nego kad se promenljive statički alociraju za vreme prevođenja. *Clipper 5.01* poznaje deklaracije **STATIC** i **LOCAL** koje deklarišu promenljive, ali za vreme prevođenja programa. Rad sa takvim promenljivim je značajno brži od rada sa **PUBLIC** ili **PRIVATE** promenljivim. Osim toga, **STATIC** i **LOCAL** promenljive se razlikuju i po oblasti važenja. **LOCAL** promenljiva važi u proceduri u kojoj je deklarisana, slično kao i **PRIVATE** promenljiva, ali za razliku od nje, nije vidljiva ni u jednoj proceduri/funkciji koja se poziva iz te procedure. I promenljiva klase **STATIC** se kreira na početku procedure i "nestaje" na kraju. Za **STATIC** promenljive važe ista pravila kao i za **LOCAL**, samo što se vrednost takvih promenljivih čuva i posle prestanka rada procedure. Primer sa slike 2 pokazuje oblasti važenja različitih klasa promenljivih. Atribut **STATIC** može se napisati i ispred deklaracije procedure ili funkcije (**STATIC PROCEDURE/FUNCTION**) i tada služi za označavanje procedure koja se može pozivati samo iz tekuće programske datoteke. Ovo može biti korisno kod pisanja lokalnih funkcija koje nećete pozivati iz ostalih delova programa i tako ne morate brinuti da ćete slučajno "pogoditi" ime te privremene funkcije, pišući ponovo neku takvu funkciju.

ON-LINE DOKUMENTACIJA

Značajnom olakšanju svakodnevnog rada svakako će doprineti i popularni *Norton Guide* – rezidentna baza podataka sa prepisanim skoro svim priručnicima za *Clipper 5.01*. Ovaj uslužni program ste verovatno već koristili. Kada se startuje, ostaje rezidentan i kasnije se startuje sa proizvoljnog mesta kombinacijom tastera koju sami odaberete. Na primer, pišete program u editoru i zaželite da saznate sintaksu neke funkcije. Umesto da listate dokumentaciju, dovoljno je pritisnuti <**Shift-F1**> i moći ćete da pretražujete "elektronski" udžbenik. Ako kursor u editoru pozicionirate na naziv funkcije koju tražite, nećete morati ni da pretražujete bazu, automatski se dobijaju podaci o funkciji na čijem je nazivu kursor. Na taj način nećete morati svakog trena da listate priručnike, već je dovoljno da pritisnete taster i da pročitate potrebnu informaciju. Najnovija verzija *Clipper*-a sadrži *Norton Guide* sa zaista svim što će vam zatrebati u radu: detaljni opisi funkcija, komandi, operatora, razlika nove od prethodnih verzija *Clipper*-a, posebne napomene itd. Jedino nedostaje *Debugger*, za koga postoji dokumentacija u obliku ASCII datoteke.



OVERLEJI, MEMORIJA, VMM

Pri pisanju malo većih aplikacija u verziji *Summer'87*, redovno se javljao problem slobodne memorije. Ako pogledate veličinu EXE datoteka koje generiše novi *Clipper*, videćete da su one uglavnom veće i to za primetnu brojku. Ipak, nova verzija, kao što ćemo videti, rešava u potpunosti taj problem. *Clipper Summer'87* je uvek po učitavanju programa u memoriju statički alocirao potrebnu memoriju – ako bi se u toku izvršavanja programa javila potreba za samo još jednim slobodnim bajtom – program je neumoljivo iskakao. Pošto je program uvek bio dug (a biće i ubuduće), svaki rad sa memo poljima i dužim stringovima je bio vrlo opasan. Preraspodela slobodnog prostora mogla se vršiti primitivnim menjanjem prostora za program i za podatke korišćenjem DOS komande **SET CLIPPER**, kao i korišćenjem overleja, ali, sve su to bila polovična rešenja iz mnogo razloga. EMS memorija je korišćena samo za smeštanje indeksa. Novi *Clipper* ima rešenja za sve ove probleme. Pristup memoriji se u novom *Clipper*-u obavlja pomoću **virtualne memorije**. To je sistem davno razrađen na "velikim" operativnim sistemima: program pristupa memoriji preko posrednika koji odzvoljava korišćenje memorije u blokovima. Ako u operativnoj memoriji ne postoji dovoljno prostora za sve potrebne blokove, koriste se pomoćne memorije za prihval blokova (EMS, disk). To usporava program, ali zato omogućava povećanje ukupnog adresnog prostora. Novi *Clipper* ima adresni prostor od čak 64MB!

VMM (*Virtual Memory Manager*), kako se zove *Clipper*-ov sistem za upravljanje memorijom, omogućava korišćenje do 64MB memorije za program, podatke i bafere za datoteke. Deo osnovne memorije i EMS memorija se koriste dok postoji slobodan prostor. Ako je potrebno još memorije, vrši se upisivanje podataka na disk (*Swapping*). To omogućuje da koristite onoliko memorije koliko imate slobodnog prostora na disku. Na ogromnu žalost, u praksi ne ide sve tako glatko. VMM radi veoma dobro i brzo na test primerima, međutim, u verziji 5.0 se vrlo često dešavalo da program "iskoči" sa nekom greškom tipa *Internal Error*, da na disku ostanu neke datoteke sa imenima poput "142453C3D" ili čak da se računar potpuno blokira. Sve ovo ukazuje da ovaj deo novog *Clipper*-a nije do kraja završen – verzija 5.01 se puno bolje ponaša, ali još uvek treba zadržati malu dozu opreza.

Mada pravi virtualni adresni prostor ukida potrebu za klasičnim overlejima, novi *Clipper* poznaje termin **dinamički overleji**. Dinamički overleji su odbro zamišljeni i trebali bi u potpunosti da zamene statičke. Naime, pri linkovanju se automatski cela aplikacija deli na overleje. Svi overleji se dodaju osnovnoj EXE datoteci, tako da se dobija jedna EXE datoteka (moguće je razdvojiti overleje i u posebne datoteke). Pri izvršavanju, ako nema dovoljno memorije, overlej menadžer učitava novi overlej po LRU algoritmu i tako smenjuje potrebnu memoriju. Programer bukvalno ne treba ništa da priprema – čak i pozive procedura iz jednog u drugi overlej rešava sam overlej menadžer. Statički overleji se više ne koriste, osim kad se linkuju procedure pisane u C-u i asembleru. Za statičkim overlejima pod ovakvim uslovima i ne postoji stvarna potreba.

Očigledno je da novi *Clipper* ima veoma kompleksan pristup memoriji, što se i te kako isplati ako se imaju u vidu prednosti. Iz ovoga se može zaključiti da velika dužina EXE datoteka koja se često uzima kao mana naročito nove verzije *Clipper*-a i nije tako velika mana kad se uzme u obzir šta sve mora da sadrži bilo koja aplikacija: VMM, menadžer dinamičkih overleja, sistem upravljanje memorijom za nizove i stringove, procedure za rad sa datotekama baze podataka... Da se nešto veće zauzeće diska isplati vidi se i iz primera programa DBU, koji se standardno isporučuje uz *Clipper*. Mada se stari i novi DBU neznatno razlikuju, obavljaju skoro identičnu funkciju. Stari DBU na disku zauzima oko 270, a novi čak oko 360K. Nekom će možda i to biti dovoljno da ostane na verziji *Summer'87*, ali prvi pogled vara. Naime, šta vam znači što je program na disku mali kad se pri učitavanju "raširi" po celoj memoriji? Ako startujete novi i stari DBU i pomoću opcije RUN utvrdite koliko memorije zauzimaju videćete sledeće podatke: stari zauzima oko 408, novi oko 363, sa jednom ogromnom razlikom – novi **nema** ograničenje na ovu maksimalnu vrednost – ako mu zatreba memorije, iskoristiće EMS ili disk, dok stari neminovno prijavljuje grešku. Za primer, testirajte program sa slike 3 koji prosto "guta" memoriju – *Clipper Summer'87* neće izdržati više od par iteracija, dok će novi raditi sve dok imate slobodnog prostora na disku.

NOVI LINKER

Pošto linker verzije *Summer'87* radi zaista vrlo sporo, obično programeri upotrebljavaju *Turbo* linker koji posao uredi za neuporedivo manje vremena. Međutim, *TLINK* ne može da kreira overleje a kod koji pravi može biti primetno sporiji od ekvivalentnog pravljenog pomoću *PLINK*-a (pokušajte da linkujete neku veću aplikaciju sa puno procedura *TLINK*-om pa onda *PLINK*-om). *Clipper 5.0* ima specijalni linker *RTLink* koji rešava dosta dosadašnjih probleme sa linkerom. Jedna novina se zove PLL biblioteke (*Pre-Link Library*) a druga *Incremental Linking*.

Ideja PLL biblioteka je slična ideji *run-time* modula: osnovna aplikacija se sastoji samo od koda koji je korisnik napisao, dok se sistemske rutine smeštaju u posebnu datoteku koju osnovna aplikacija učitava za vreme izvršavanja po potrebi. Na taj način se drastično smanjuje veličina programa na disku (možete li da zamislite *Clipper* program od 4K?) i vreme linkovanja. Za vreme izvršavanja programa, potrebni moduli se učitavaju iz takozvane PLL biblioteke, tako da program u memoriji nije manji – manji je samo na disku. Pri instalaciji *Clipper 5.0* kreira BASE50 biblioteku koja sadrži sistemske module za svaki *Clipper* program. Ako pri linkovanju navedete opciju PLL BASE50, vaš EXE program će biti manji za oko 130K, a pri izvršavanju će se potrebni moduli pozivati iz biblioteke BASE50. PLL biblioteku možete i sami da kreirate smeštanjem opštih procedura koje često koristite i tako smanjite zauzet prostor na disku i skratite vreme linkovanja. Kad završite razvoj programa, preporučljivo je da napravite završnu aplikaciju bez PLL biblioteke, zato što će vam tako aplikacija biti "u komadu" i zato što je rad sa PLL bibliotekama predviđen samo za razvoj.

Ideja inkrementalnog linkovanja nije nova – *Microsoft* linkeri su imali još pre nekoliko godina ovu mogućnost. Radi se o tome da se pri linkovanju vrši provera da li je modul menjan i da se oni koji nisu menjani ne povezuju ponovo, što rezultuje bržim linkovanjem. Međutim, ova mogućnost u praksi ne pokazuje naročite rezultate i dosta često rezultuje nekom greškom *RTLink*-a koje nema u spisku grešaka (možda se radi o nekoj nedozvoljenoj kombinaciji prekidača – ko zna). U dokumentaciji piše da ovu mogućnost treba koristiti kod "velikih" aplikacija – kod malih se jedva neće primetiti.

Pošto novi *Clipper* ima specifične dinamičke overleje, vrlo je mudro ne koristiti neki drugi linker već baš njegov *RTLink* – PLL opcije i specificiranje RAM diska kao privremenog kataloga mogu učiniti vreme čekanja sasvim prihvatljivim, a rezultat je mnogo pouzdaniji. Uostalom, ne treba zaboraviti da je *RTLink* pisan specijalno za *Clipper*, pa je verovatno napravljen tako da izvuče iz njega maksimum.

DEBUGGER

Verzija *Summer'87* je imala *debugger* za kojim su svi posezali samo u krajnjoj nuždi – bez obzira što je omogućavao pregled promenljivih, datoteka, postavljanje prekidnih tačaka itd., ispravljanje programa "na slepo" nije baš najugodnija disciplina, a stalni odlazak u editor zbog pregledanja teksta programa može biti veoma razdražujući. Nova verzija ima značajno poboljšanje: *source level debugger*. *Debugger* je vrlo sličan *Microsoft*-ovom *CodeView debugger*-u. Većina komandi se može dobiti iz menija ili kucanjem odgovarajuće komande. Postoje sve nužne komande: pregled listinga, praćenje vrednosti promenljivih, postavljanje prekidnih tačaka, pregled tekućeg stanja, itd. Moguće je videti i listing programa posle rada pretprocesora (tada treba prevesti program sa opcijom /P). Što se tiče skupa komandi, moglo bi se ponešto i dodati, ali spisak komandi je ipak zadovoljavajući. Može se, kao u prethodnoj verziji, ulinkovati u aplikaciju a moguće je da se i posebno startuje iz komandne linije. *Debugger* predstavlja ogromnu pomoć za ispravljanje grešaka. Na žalost, i ovaj deo nije

```
Clipper 5.01a » The Guide To Clipper » Language » Functions
 Expand   Search      Options    Language    Tables

DAY()            Return the day of the month as a numeric value
DBAPPEND()       Add a new record
DBCLEARFIL()     Clear a filter condition
DBCLEARIND()     Close indexes
DBCLEARREL()     Clear active relations
DBCLOSEALL()     Close all occupied work areas
DBCLOSEAREA()    Close a work area
DBCOMMIT()       Flush pending updates

                    }

CBlk := {|x| x*=2  Func(x) }         // Code block

REPEAT
   HELP "Ovo je help linija"
   Izbor := Menu ("M1"  opc  izbor++)
   DO CASE
      CASE Izbor == 0, EXIT
      CASE Izbor == 1, Func1(CBlk), LOOP
      CASE Izbor == 2; Func2(), LOOP
      CASE Izbor == 3, Func3(), EXIT
BRIEF u3 0 - (c) 1984-1990 by UnderWare  Inc     Line  24   Col  1    †# 12 29 am
```

previše pouzdan jer dosta često blokira računar, naročito kod komlikovanijeg koda. Verzija 5.01 je donela potpuno novi *Debugger* koji ima dosta novih mogućnosti i izgleda mnogo pouzdanije.

OBJEKTNO PROGRAMIRANJE I GET SISTEM

Novi GET sistem omogućava izradu maski za unos podataka kao i stari, ali ima potpuno novu strukturu i u suštini je mnogo fleksibilniji. Kao novost u GET naredbi dodate su opcije WHEN i COLOR. Opcijom WHEN se zadaje uslov koji proverava ispravnost unete vrednosti slično kao VALID, ali pre unosa vrednosti promenljive. Ako uslov nije ispunjen promenljiva se preskače i vrši se unos sledeće u spisku. COLOR omogućava nezavisnu kontrolu boje svake promenljive. Bez obzira na korist od ovih opcija, prava novina je u arhitekturi ovog sistema. Naime, *Clipper 5.0* sasvim neočekivano ima elemente objektnog programskog jezika (*OOP*). Detaljnije objašnjenje filozofije objektnog programiranja zahteva malo veći prostor – to je sasvim posebna tema. Bez dubljeg ulaženja u suštinu, radi se o postojanju specijalnog tipa podataka (*objekti*) i operacija sa takvim tipom. Korišćenjem specijalnih objekata izrađeni su *Error*, *Get* i *Browse* sistemi koji omogućuju veliku fleksibilnost uz vrlo jednostavno programiranje standardnih stvari. Tako je ceo GET sistem realizovan kao dosta jednostavan *Clipper* program (sa datim tekstom programa) i pri tome su podržane sve mogućnosti starog sistema i nove WHEN i COLOR opcije. Kako sve to radi? Svaka @ SAY GET naredba kreira jedan objekat koji sadrži sve podatke o promenljivoj koja se učitava: koordinatama ispisa, bojama ispisa, itd. i referenca tog objekta se upisuje u globalni niz **GetList**. Naredbu **READ** pretprocesor prevodi u poziv funkcije **ReadModal** (i za nju postoji tekst *Clipper* programa) koja realizuje standardno editovanje navedenih promenljivih. Šta je dobijeno? Ako koristite stari skup SAY GET naredbi, osim WHEN i COLOR opcija, ništa! Ali, ako vam zatreba neka specifična maska za unos, stoji vam na raspolaganju potpuna kontrola GET objekata pomoću koje možete napraviti zaista skoro sve što vam ikad zatreba. Najjednostavnija (a preko potrebna) stvar je realizacija READ naredbi unutar druge READ naredbe. Ako se reference novih GET objekata upisuju u lokalni niz **GetList**, stari **GetList** će ostati netaknut. Primer jednostavne upotreba READ GET naredbi vidite na slici 4. Posle učitavanja imena i prezimena, ako stranka radi uzimaju se dodatni podaci o adresi radne organizacije posebnim GET READ naredbama. Zatim se učitava zanimanje stranke, ali samo ako stranka radi. Na slici 5 je prikazan primer učitavanja jedne promenljive, ali na nižem nivou, korišćenjem operacija sa objektima.

Poboljšanju korisničkog interfejsa treba dodati i zgodnu funkciju **Alert** koja realizuje izbor između zadatih opcija u malom prozoru. Na primer poziv

Alert("Izaberite opciju..",{"Printer","Datoteka","Izlaz"})

ispisuje simpatičan prozor u kome birate jednu od ponuđenih opcija.

BROWSE SISTEM

Prodor objektnog programiranja u *Clipper* ovim nije završen: svi koji znaju velike mogućnosti **DBEdit** funkcije iznenadiće se kad vide da je te funkcija označena kao "funkcija koja postoji zbog kompatibilnosti sa starijim verzijama i ne preporučuje se za upotrebu". Alternativa **DBEdit** funkciji koja se nudi su *TBrowse* i *TBColumn* objekti. Pre nego što objasnimo o čemu se radi, zadržimo se za trenutak na prirodnosti ideje objektnog programiranja. Ako imate malo više iskustva u programiranju na *Clipper* u, svakako ste koristili funkciju **DBEdit**. Otvorite datoteku, definišete hedere, sadržaj kolona, itd., pozovete funkciju **DBEdit** i na ekranu korisnik može da razgleda tabalarno date podatke iz datoteke. I to je savršeno rešenje – dok vam ne zatreba **još samo** da "kad korisnik pritisne F5, da se uradi..." Rešenje je pronađeno u naredbi **SET KEY** pomoću koje ste mogli da specificirate ime funkcije koja bi realizovala potrebnu operaciju. Tako ste **DBEdit** funkciji **saopštavali** da želite da se, kad korisnik pritisne taster F5, pozove određena funkcija. Međutim, onda se ukazala potreba da i **DBEdit** funkcija vama odgovori na neki način (na primer, da vas **obavesti** o kodu tastera koji je pritisnut). Dalje, ponekad je potrebno da **naložite DBEdit** funkciji da nešto uradi, na primer, ako ste menjali tekući slog da regeneriše sliku na ekranu, itd. – kao što vidite, zahtevima nikad kraja. Pošto je ovako nešto vrlo česta potreba, **DBEdit** funkcija ima mogućnost zadavanja imena funkcije koja se poziva uvek kad korisnik pritisne neki taster. Na taj način možete da **komunicirate** sa funkcijom **DBEdit**: ona vama, na primer, pošalje poruku "pritisnut je taster ENTER", a zatim vaša funkcija odgovara "regeneriši ekran" i sl.

Komunikacija se obavljala korišćenjem specijalnih kodova. I mada je to dosta dobro radilo, ipak bi pre ili kasnije došlo do nekog ograničenja, jer je taj sistem nestabilan po prirodi: sa jedne strane je dobro da **DBEdit** bude "crna kutija" i da se vi ne bavite njegovom strukturom, ali sa druge strane ponekad je potrebno da utičete veoma suptilno na neke akcije te "crne kutije". Objektno programiranje savršeno prirodno rešava taj problem. Ako nemate iskustva sa objektnim programiranjem, ne bojte se: *Clipper* sadrži samo neke elemente za rad sa objektima, sa kojima ćete se vrlo brzo familijarizovati i kasnije nećete moći da verujete da ste ranije programirali bez njih.

DBEdit se zamenjuje prirodnom reprezentacijom: sa jedne strane je objekat koji služi za tabalarni pregled datoteka (tzv. *TBrowse* objekat) a sa druge strane vaša procedura koja komunicira sa njim. Sada ne preko indikatora, već korišćenjem specijalnih operatora za **slanje i primanja poruka**. Na početku je potrebno kreirati objekat koji će vršiti pregled. To radimo pozivom funkcije *TBrowseDB*. Svaki takav *TBrowse* objekat sadrži više *TBColumn* objekata a svaki od njih opisuje jednu kolonu interaktivnog tabelarnog prikaza. Na primeru sa slike 6 prvo kreiramo *TBColumn* objekte za svaku od kolona prikaza funkcijom *TBColumnNew*, pri čemu inicijalno označavamo šta će se prikazivati u datoj koloni i kakvo će biti zaglavlje. Raznim operacijama "podešavamo" svaku od kolona, a zatim upravljamo kreiranim *TBrowse* objektom tako što mu na osnovu pritisnutog tastera šaljemo "poruke" o tome šta treba da uradi: da pomeri kursor, promeni boju i slično. Primetite da je moguće prekinuti ispisivanje tabele i izvršiti neku akciju, pa tek onda nastaviti ispisivanja. Opisani princip omogućava potpunu kontrolu i vrlo jednostavno programiranje. Kao što smo već videli, slična koncepcija se primenjivala i kod **DBEdit** funkcije, samo je ovde mnogo generalnija i fleksibilnija. Primer sa slike 6 omogućava jednostavnu zamenu tukcije **DBEdit**.

Ovim se ne iscrpljuju prednosti novog *TBrowse* sistema. Naime, ne postoje nikakve prepreke da vršite tabalarni pregled **bilo** kog tipa podataka – na primer dvodimenzionalnih nizova.

PRIHVATITI ILI NE

Kao što vidite, novi *Clipper* ima puno novina, od kojih su neke zaista izuzetne. Pomak ka C-u (uslovno rečeno) odgovara profesionalnim programerima, dok ostalima ne smeta. Kodni blokovi, nove funkcije i naredbe, objekti, pretprocesor, virtualna memorija i *debugger* zaokružuju celinu koja može biti odličan sistem za razvoj manjih baza podataka. Možda je trebalo malo više pažnje posvetiti nivou baza podataka: nekoliko novih naredbi za upite, postavljenje relacija, itd., odnosno pomak ka "velikim bazama" svakako ne bi škodio. Ne treba zaboraviti da je novi *Clipper* u svim kritičnim delovima značajno brži: na primer, operacije sa **LOCAL** tipom promenljivih su više od dva puta brže od ekvivalentnih operacija sa **PRIVATE** promenljivim. Ipak, najveći problem je pouzdanost svega toga – šta vam vrede sve nove komande, ako program nekontrolisano iskače. Pošto je *Clipper* iz korena izmenjen, donekle je razumljivo što se javio toliki broj grešaka u verziji 5.0. Verzija 5.01 se ponaša mnogo sigurnije, ali sigurna ocena pouzdanosti može se dati tek posle nekoliko meseci korišćenja.

Međutim, može se zaključiti da je novi *Clipper* svakako vredan pažnje – očigledno izuzetno koncipiran, on nudi ogromne mogućnosti. Uz sve treba imati u vidu da su mu šanse na uspeh višestruko povećane izuzetnom rasprostranjenošću verzije *Summer '87*.

DOSTOJAN SVOG IMENA

Baze podataka koje rade u *Windows* okruženju tek počinju da se afirmiraju na tržištu. Ovoga puta predstavljamo vodeći softverski paket ove vrste. Poslije upoznavanja sa *Superbase 4*, nužno nam se nameće zaključak: vladavina baza podataka koje rade u tekst modu polako ali sigurno prestaje.

Zoran Pletikosa

Ako pogledamo unazad par godina, vidjećemo da razvoj softvera prati napredak u brzini procesiranja i tehnologiji ekranskog prikaza. Prve baze podataka su nastale u vrijeme kada nije ni bilo pravih grafičkih kartica, a sljedeće njihove verzije su morale sačuvati osnovnu koncepciju. Naravno, kod ove vrste programa postojanje grafike na računaru nije ni bilo od presudnog značaja, jer su se tekstualni i numerički podaci mogli unositi, sortirati i pretraživati i na najobičnijem alfanumeričkom ekranu. Ali podatke je trebalo i odštampati na određen način i u određenom obliku. Kada smo svojevremeno radili sa dBASE III+, najveći nedostatak nam je bio vrlo ograničena mogućnost za oblikovanje izvještaja u interaktivnom načinu, bez upuštanja u programiranje, a ako bi se neko odlučio i na ovo posljednje, rezultat na printeru bi bio daleko od optimalnog. Kasnije su se pojavili programi nezavisnih kuća koji su pokušavali ispraviti ove nedostatke i koji su više ili manje uspješno štampali DBF datoteke u obliku maski, ali sve je to bilo samo privremeno rješenje.

KO PRE PROZORU...

Imali smo, nadalje, prilike vidjeti za kakvim trikovima programeri, uglavnom u *Clipper*-u, posežu kada je trebalo u datoteku uključiti neku sliku, ili na istoj strani sa tekstualnim podacima odštampati neki zaštitni znak, amblem ili nešto slično. Meniji u aplikacijama su bili veoma cijenjena tema među programerima, svaki od njih nas je pokušavao ubijediti da je njegovo rješenje najbrže i najbolje. I tako bi to i ostalo da se nije pojavio *Windows* koji je u verziji 3.0, poslije preboljelih dječijih bolesti, na najboljem putu da postane standard za grafičke operativne sisteme. I ranije je bilo pokušaja od strane Digital Research-a da svoj GEM pokuša progurati na tržištu, ali bez znatnijeg uspjeha, ne zbog toga što je proizvod bio loš, već zato što mu je nedostajao blagoslov jednog Microsofta i IBM-a. A u to vrijeme je Microsoft već razvijao konkurentski proizvod.

Sada se softverske kuće prosto utrkuju u izbacivanju programa koji rade u *Windows* okruženju. Budući da se tržište ove vrste softvera tek formira, razumljivo je nastojanje proizvođača da baš njihov proizvod bude taj koji će postaviti temelje za određenu kategoriju programa i u budućnosti diktirati standarde koje konkurenti moraju da poštuju. Istorija je pokazala da standarde postavlja onaj ko reaguje u pravo vrijeme i popuni prazninu na tržištu. U pojedinim kategorijama *Windows* softvera već sada se situacija pomalo izbistrava, pa kod tekst procesora bitku biju Microsoft *WinWord* i Lotusov *Ami Professional*, za prvo mjesto kod tabelarnih proračuna utrkuju se *Microsoft Excel* i *Informix WingZ*, a što se tiče baza podataka za sada je *Superbase 4* neprikosnoven, mada i on ima konkurenciju. U zadnje vrijeme se pominje *dbVISTA III* za *Windows* koju, nažalost, do sada nismo imali prilike da vidimo.

I tako sve je više softvera koji radi pod prozorima. S druge strane, *Windows*, pa samim tim i programi koji rade pod njim imaju i određenih nedostataka, ali ipak smo dojma da se radi o pravom putu i ako krenete njim sigurno ne možete pogriješiti.

PAKET, INSTALACIJA, LITERATURA

Superbase 4 for Windows je proizvod engleske softverske kuće Precision Software. Program dolazi u povelikoj, izuzetno lijepo dizajniranoj kartonskoj kutiji koja je ipak suviše njezna za naš ukus. Već sama težina paketa od 4 kg je impresivna, sadržaj takođe. Unutra ćete naći dvije 5.25 inčne diskete od 1,2 MB, četiri 3.5 inčne diskete od 720 KB, dva kožna registratora fascikla u koje trebate da uložite literaturu, brižljivo u celofan zapakovana uputstva na preko 900 stranica, knjižicu koja objašnjava primjere i registracionu karticu. Odmah se da primijetiti da nema ni govora o recimo 10 ili više disketa kao što se isporučuju neki drugi softverski paketi, već samo o dvije HD diskete od kojih su na jednoj programi, a na drugoj arhivirani primjeri. Mali broj disketa je inače odlika programa koji rade u *Windows* okruženju, jer nije potrebno posebno isporučivati pobudu za more grafičkih kartica i printera. To je na sebe preuzeo sam *Windows*.

Ukoliko je vaš računar opremljen sa 3.5 inčnom disketnom jedinicom, nema nikakvih problema da nju iskoristite za instalaciju. Za ovaj potez Precision Software zaslužuje pohvale jer se inače lako može desiti da, kada kupujete poštom neki program, dobijete isti samo na 5,25 inčnim disketama, uz ljubaznu napomenu da su na raspolaganju i 3,5 inčne diskete ali na poseban zahtjev! Zbog toga je poželjno uvijek pri naručivanju naznačiti na kakvim disketama želite da vam se program isporuči. Ipak lijepo je vidjeti da ima proizvođača koji misle o nama korisnicima i ne nastoje po svaku cijenu da uštede dolar-dva na disketama kod programa koji koštaju 200 puta više.

Instalaciju na hard disk obavlja priloženi SETUP program sa programske diskete. Od vas se samo traži da naznačite ciljni direktorij i da li želite da se instaliraju i primjeri. Poželjno je i njih instalirati bar za prvi period dok ne uđete u fazon, jer su dosta indikativni, a kasnije ih lako možete obrisati. Procedura instalacije na našem 12 MHz AT-u sa 28 ms hard diskom je trajala oko 3 minuta i skratila je hard disk za 3 Mb formirajući pored osnovnog još šest subdirektorija za brojne primjere. Interesantno je napomenuti da sam izvršni program i prateće datoteke potrebne za rad zauzimaju 966 Kb (ukupno 7 datoteka), a sav ostatak do 3 Mb zauzima ukupno 270 datoteka primjera! Dakle, kasnije kada primjere obrišete, zauzeće diska će dosta biti skromno za program takve klase.

Kada prvi puta startate program, naravno iz *Windows*-a, bićete zamoljeni da unesete svoje cijenjeno ime i instituciju u koju ste udružili svoj rad. Ovo posljednje smo morali unijeti, iako smo se svojski trudili da dokažemo programu da se nalazimo u radnoj sobi a ne na radnom mjestu. Nije pomoglo. Ova dva podatka će se pojavljivati na uvodnom ekranu pri svakom sljedećem startovanju programa. Zatim morate unijeti serijski broj vaše kopije. Naravno, pokušali smo da mu podvalimo nekoliko brojeva koji su nam prvi pali na pamet da vidimo kako će reagovati. Od nas se zahtijevalo da umetnemo Program Disk u disketnu jedinicu, i odmah smo pozdravljeni porukom da broj nije korektan i da pokušamo ponovo. Ovoga puta smo ukucali tačan broj i sve je bilo u redu.

Prema tome, Precision Software nije zaštitio program od kopiranja (čak se preporučuje da napravite radne kopije oba diska), već je posegnuo za jednim drugim vidom zaštite koji je sve više u modi. Naime, niko vas ne sprečava da dijelite kopije disketa zajedno sa serijskim brojem, ali baš na osnovu tog broja i registracione kartice koju ste poslali proizvođaču vrlo lako je utvrditi koji je korisnik pustio program u opticaj. Zato pamet u glavu, jer bi vam u protivnom mogao pozvoniti sudski pozivar na vratima jednog lijepog dana.

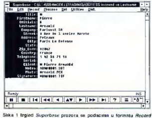
Uputstva dolaze zapakovana u celofan i nepovezana tako da ćete morati sami da ih složite u dva fascikla. Moramo priznati da nam se ovaj način opreme literature ne sviđa previše, jer su fascikle glomazne i nezgodne pri čitanju. S druge strane, kada ih uložite u registrator, vrlo će lijepo izgledati na polici. O ukusima se ne raspravlja, ali pri ruci imamo literaturu koja se isporučuje uz *NeuralWorks Professional II/Plus*, ona je u formi pravih knjiga uvezanih plastičnim oprugama i mnogo su zgodnije i za nošenje i za čitanje.

Papir je izuzetno kvalitetan, čak imate i plastične granične listove između pojedinih cjelina da lakše nađete ono što želite. Grafička oprema je na najvišem nivou, ali ostaje nejasno zašto tekst nije poravnat sa obje strane, već ovako imate dojam da je kucan na pisaćoj mašini. Ovo jeste estetski detalj, ali sada kada se i tekst pisama uravnava, za ovaj propust teško da se može naći opravdanje. Međutim, uputstva za mnoge druge komercijalne programe su odštampana na isti način pa ovo nije izuzetak.

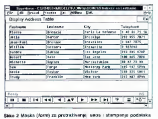
Dokumentacija je pisana izuzetno razumljivim i čitljivim stilom, dosta je detaljna i opremljena dobrim indeksom pojmova. Jedino redoslijed izlaganja bi mogao biti malo drugačiji, npr. poglavlje „Learning to use Superbase 4" se nalazi na kraju prve knjige, a ne na početku kao što bi se očekivalo. Takođe ne bi škodilo malo više slika ekrana i primjera u tekstu. Nije da ih nema, ali kada čitate o gradnji i upotrebi **Query** upita, vrlo se lako izgubite u masi teksta. Sve u svemu, literatura je svakako primjerena programu ove klase.

PRESS PLAY

Korisnicima Atari ST računara je vjerovatno poznat ovakav sličan program iste kuće pod nazivom *Superbase Professional*. Ukoliko se



Slika 1 Izgled *Superbase* prozora sa podacima u formatu *Record View*. Poslednja tri polja su eksterna.



Slika 2 Maska (*form*) za pretraživanje, unos i stampanje podataka



Slika 3 Glavna novost – uključivanje grafike u bazu podataka



Slika 4 *Form Designer* – radna površina sa alatima (toolbox), lenjirom (*ruler*) i mrezom (*grid*)

situacije u međuvremenu nije promijenila radi se o najcjenjenijoj bazi podataka za ST računare. Budući da je teško praviti revolucionarne poteze u razvoju bilo koje kategorije softvera pa samim tim i baza podataka. Superbase je uveo jednu dosta intreresantnu novinu, više evolucionog karaktera a to je poseban način za pretraživanje baze podataka koji nastoji oponašati funkcije video rikordera. Tako na dnu ekrana nalazite 12 stilizovanih tastera za manipulisanje datotekom koji se mogu aktivirati mišem ili odgovarajućim ekvivalentima sa tastature. Postoje komande za „premotavanje" zapisa pojedinačno u oba smjera (to je PLAY), ubrzano naprijed i nazad, pauza, stop, skok na početak, skok na kraj datoteke tzv. LOOKUP taster (?) za nalaženje zapisa na osnovu vrijednosti indeksiranog polja taster za formiranje filtera (=) i „kamera" komanda za prikazivanje eksternih polja sa diska na radnoj površini.

SLIKE U BAZI

Sljedeća novost koju je uveo *Superbase 4* je veoma lako uključivanje grafike u bazu podataka. Nema nikakve prepreke da jedno polje pri definiranju baze podataka označite kao eksterno, a u njega kasnije upisujete samo naziv datoteke sa slikom praćene odgovarajućim putem za njeno pronalaženje na disku. Pri pregledavanju baze ovi grafički podaci se neće inicijalno prikazivati na ekranu, već je potrebno aktivirati kameru da bi se otvorio poseban prozor. On se po potrebi može pomjerati, povećavati i na kraju ga zatvorite kao bilo koji drugi *Windows* prozor. Ukoliko u zapisu (**record**) ima više eksternih polja, oni se mogu sukcesivno pozivati na ekran sa NEW EXTERNAL opcijom iz RECORD menija. Opcija PREVIOUS EXTERNAL radi isto u suprotnom smjeru.

Podržani su sljedeći grafički formati

- **IMG** – *Gem Paint* bit mapirana slika
- **PCX** – *PC Paintbrush* bit mapirana slika
- **TIFF** – skeneri
- **EPS** – *Encapsulated Postscript*
- **BMP** – *Windows* 3.0 ekranski nezavisni bit map format
- **WMF** – *Windows* metafile format

Ove eksterne datoteke se veoma lako uključuju u ulazno-izlaznu masku (**Form**). Neophodno je jedino na maski označiti okvir gdje treba da se uklopi slika i dodijeliti mu naziv polja sa eksternom datotekom. Slike će se skalirati automatski na zadatu veličinu. S obzirom da bit mapirana grafika, naročito ako je u dosta boja, zauzima mnogo RAM memorije kada se učita sa diska predviđen je i prekidač „s" koji obezbjeđuje skaliranje za vrijeme učitavenja a ne kasnije i time dosta uštedi u memoriji. Takođe postoje prekidači koji omogućavaju da se očuva tzv. ASPECT RATIO tj. odnos vertikalne i horizontalne dimenzije slike („a") i prekidač koji omogućava bolje prilagođavanje slike u koloru koja će se prikazati na monohromatskom ekranu („m").

Na primjer u polje pod nazivom SLIKA možete unijeti vrijednost „Ptica.pcx s" (navodnici su u ovom slučaju neophodni).

Iako to ne zavisi od *Superbase*-a, nedostatak cijelog ovog koncepta je u izuzetno velikoj količini memorije koju slike znaju zauzeti na hard disku, pa pregledavanje datoteke sa prikazivanjem teksta i grafike istovremeno se zna pretvoriti u pravu muku za hard disk i onog koji to mora raditi. Dakle, poželjan je što brži računar i što brži disk.

KREIRANJE BAZE

Definiranje karakteristika vaše datokete u principu je slično onome u dBASE III+, ali je u *Superbase*-u dovedeno skoro do savršenstva. Odmah na početku se srećete sa još jednom veoma važnom novinom, mogućnošću kreiranja različitih nivoa šifara za zaštitu podataka. Prvi nivo omogućava puni pristup podacima, uključujući i mogućnost brisanja. Drugi novo omogućava korisnicima samo čitanje i pisanje, dok treći nivo znači da korisnici mogu samo ▶

čitati podatke a ne mogu ih brisati, mijenjati ili ažurirati. Kada postavite šifre, sljedeći puta kada otvorite datoteku moraćete da ih unesete prije početka rada. Podrazumijeva se da se prilikom unošenja šifre tekst ne pojavljuje na ekranu. Pazite da ne zaboravite šifru jer se iste može modifikovati kasnije samo koristeći DML komandu PASSWORD.

Superbase poznaje tri osnovna tipa polja (**Field**): **Text/Logical, Numeric** i **Date/Time**, a pri preciznijem određivanju njihovih karakteristika proći ćete kroz niz okvira za dijalog sa čitavim nizom mogućih i nemogućih formata. Polja mogu imati niz dodatnih atributa kao:

- **REQUIRED** (ne može ostati prazno pri unosu podataka)
- **READ ONLY** (ne mogu se unositi podaci u njega)
- **CALCULATION** (sadrži formulu koja izračunava vrijednost iz drugih polja i uključuje je u to polje)
- **CONSTANT** (veoma slično kao prethodno, ali izračuna vrijednost samo jednom pri unosu podataka u zapis)
- **VALIDATION** (sadrži i formulu koja omogućava neposrednu kontrolu nad unijetim podacima npr. ŠIFRA >= 100 će javiti grešku ako uneste broj koji ne zadovoljava taj uslov. Možete čak specificirati koji tekst poruke će se u ovom slučaju pojaviti na ekranu da bi korisnik znao u čemu je pogriješio npr.

  ŠIFRA >= 100 ELSE „Šifra mora biti trocifren broj!"

  Izuzetno fleksibilno i efikasno)
- **EXTERNAL** (u ovo polje upisujemo naziv tekstualne ili grafičke datoteke što omogućava da se u bazu uključe slike ili da pojedina polja budu mnogo duže no što se na klasičan način može postići, mnogo zgodnije nego MEMO polja u dBASE III+)

Poslije kreiranja karakteristika vaše baze, morate odabrati jedno ili više polja za indeksiranje. Za razliku od dBASE III+, ovdje datoteku morate indeksirati po bar jednom polju, jer zapisi nisu inicijalno određeni svojim brojem u redoslijedu unošenja. Ovaj uslov je autoru vjerovatno potpuno razumljiv, ali korisnicima to ponekad zna i da zasmeta. Naime, neko će htjeti da redoslijed podataka bude tačno onakav kao što su se oni unosili. S otvorenim indeksom to neće biti moguće, jer ćete uvijek dobili sortirane zapise po vrijednosti indeksiranog polja. Naravno da se to može postići pomalo zaobilaznim putem tako da jedno polje označite atributom **Constant**, na njega vezete formulu za automatsko dodavanje rednog broja i iskoristite ga za indeksiranje, ali da li je ovako moralo biti?

Ukupno možete imati 999 indeksa po jednoj datoteci, ali ne treba pretjerivati jer se svi indeksi ažuriraju sa svakim novim zapisom, te ko da veći broj indeksa zahtijeva i više vremena za taj posao.

Još jedna novost je tzv. UNIQUE indeks koji onemogućava da u indeksirano polje upišete vrijednost koja je ranije već jednom bila unesena. Vrlo praktično ako se trebaju unositi npr. nekakve šifre koje uvijek moraju biti različite, sve s ciljem da se spriječi unos netačnih podataka.

Za razliku od dBASE III+, Superbase definiciju baze podataka čuva u posebnoj datoteci, a same podatke u posebnoj. Standardne ekstenzije datoteka su:

- **SBD** – definicija baze koja sadrži informacije o tipovima i formatima polja
- **SBF** – datoteka sa podacima
- **X** – gdje je x broj od 1-999. To su indeksne datoteke
- **SBV** – datoteka ulazno-izlaznih maski – **Form** kreiranih u *Form Designeru*
- **SBP** – datoteka sa programom u DML jeziku, može se izvršiti RUN komandom u DML meniju
- **SBT** – datoteka tekst editora koji je pridodat *Superbase*-u
- **SBQ** – **Query** datoteka sadrži upite za složenije pretraživanja baze
- **SBU** – **Update** datoteka, nije isto kao u dBASE-u jer ova služi za globalno mijenjanje vrijednosti pojedinih polja u okviru cijele baze
- **SBB** – formati o „naljepnicama" kao LBL u dBASE-u
- **SBK** – definicija funkcijskih tastera
- **SBM** – makro datoteka, u nju se snimaju pritisci na tastere i sva manipulacija mišem, i kasnije se one može izvršiti

RADNI AMBIJENT

Rad sa bazom podataka se obavlja u uobičajenom grafičkom *Windows* okruženju uz pomoć roletnih menija. Postoji sedam osnovnih menija:

- **FILE** (otvaranje i zatvaranje datoteka sa podacima, indeksima i maskama)
- **EDIT** (mijenjanje zapisa uz nezaobilazne **Cut**, **Copy** i **Paste** opcije)
- **RECORD** (manipulacija zapisima, unošenje, snimanje, štampanje itd.)
- **PROCESS** (najobimniji meni, obuhvata upravljanje **Query** i **Update** datotekama, štampanje izvještaja, uvoz i izvoz podataka, kreiranje naljepnica itd.)
- **SET** (postavljanje parametara za ekranski prikaz, podešavanje štampača, formata strane, definisanje funkcijskih tastera itd.)
- **UTILITIES** (nudi ekvivalente osnovnih DOS komandi, poziv tekst editora, komunikacionog modula, snimanje i aktiviranje makro komandi itd.)
- **DML** (omogućava rad sa ugrađenim programskim jezikom *Database management language*)

Superbase nudi četiri načina za formatiranje prikaza vaših podataka koji se postavljaju iz SET menija:

- **RECORD VIEW**. To je podrazumijevana opcija. Svako polje zauzima jednu ekransku liniju, nalaze se jedno ispod drugog, a na početku su inverzno data njihova imena. Ukoliko ne mogu svi podaci stati na ekran, morate upotrijebiti miša i klizače za pomjeranje ili ekvivalente sa tastature SHIFT+PGUP i SHIFT+PGDN za pomjeranje gore i dolje, te CTRL+PGUP i CTRL+PGDN za pomjeranje lijevo i desno za po jedan ekran.
- **TABLE VIEW**. Ovim se nazivi polja postavljaju po vrhu radne površine, a njihov sadržaj u kolone ispod njih. Na taj način dobijate prikaz kao u tabelarnim kalkulatorima, ali sa mnogo skromnijom matricom od 273×266.
- **PAGE VIEW**. Ovaj način omogućava da sami po svojim željama razmjestite polja na ekranu uz pomoć miša, te tako možete imati i više polja u jednoj ekranskoj liniji.
- **FORM**. Podrazumijeva da ste masku prethodno definirali u pratećem programu *Form Designer*. Maske ne nameću gotovo nikakva ograničenja, skoro sve je dopušteno: miješanje teksta, grafike, slika itd. Krajnji rezultat zavisi jedino od vašeg umjetničkog talenta.

U vezi sa prva tri ekranska formata jedna stvar nam se nikako ne dopada. Naime, ako je tekst u nekom polju duži no što može stati na ekran, ostatak nećete moći vidjeti bez skrolovanja. Nije predviđena mogućnost da tekst pređe u naredne redove, tako da pregledavanje podataka sa ovako postavljenim prikazom nije ni malo prijatno. Interesantno da kada pokušate editovati ovakvo polje, otvori se nekakav poluprozor koji preklapa druga polja i u kome je sav tekst vidljiv a njegovo mijenjanje veoma lako. Problem je što je i izlaz na štampač potpuno isti tako da sve što izlazi izvan granica ekrana, neće se odštampati na A4 papiru. Razlog ovoga je u samoj koncepciji formata – jedno polje po jednoj ekranskoj liniji. Ako dođete u ovakvu situaciju, moraćete kreirati ulaznu masku gdje je moguće slobodno odrediti dimenzije i položaj okvira za tekst svakog pojedinog polja.

PRETRAŽIVANJE PODATAKA

Kada govori o pretraživanju podataka, dokumentacija upotrebljava standardni termin BROWSE. Pogledali smo u naš Englesko-srpskohrvatski rječnik u izdanju Svjetlost Sarajevo i pronašli prevod „brstiti". Tek da se zna.

Za tu namjenu koriste se tzv. „VCR-like" komande na dnu ekrana koje su doste simpatično rješenje. Trebalo se samo dosjetiti. Pomoću njih ćete podatke „premotavati" u željenom smjeru. Pronalaženje određenog zapisa na osnovu vrijednosti nekog polja je moguće na dva načina:

- **LOOKUP** tasterom (?) koji omogućava pronalaženje zapisa na osnovu poznate vrijednosti indeksiranog polja i
- **FILTEROM** (=) koji je izuzetno eksploatisan u *Superbase*-u. Jako mnogo opcija ima filter dijalog koji dozvoljava da kreirate uslov za prikazivanje pojedinih zapisa. Koncepcija je slična onoj u dBASE-u uz uvođenje niza novina. Jedna je tzv. „LIKE" operator koji omogućava pretragu na osnovu šablona. U okviru ovog uslova je dozvoljen i čitav niz džoker znakova. Npr. ako tražite osobu kojoj ste zaboravili ime, ali znate da počinje sa slovom M, možete u filtar ugraditi komandu IME LIKE „M*", a nebitno je uopšte da li su slova velika ili male. Jedini nedostatak ove jako fleksibilne opcije je u mnogo dužem vremenu izvršavanja od **Lookup** komande, ili pak ako se u filteru upotrebljava samo znak jednakosti.

Sumirano, pretraživanje podataka u *Superbase*-u nije loše, ali bi moglo biti i malo bolje

OSNOVNE OSOBINE

Polja po zapisu i veličina zapisa	Ograničeni memorijom
Zapisa po datoteci	Ograničeno veličinom diska (teoretski do 16 miliona)
Indeksa po datoteci	maksimalno 999, svi mogu biti otvoreni
Veličine datoteke	Ograničene DOS-om i veličinom diska (teoretski 17 Gb)
Broj otvorenih datoteka	Neograničen
Strana u okviru jedne maske	Ograničeno memorijom
Maski po datoteci	Neograničeno
Tipovi brojeva	Integer, Real i Long (real do 14 značajnih cifara)
Maksimalna dužina polja	4000 karaktera, dozvoljeni i pritisci na Enter
Zaštita podataka	Tri nivoa šifara za ograničavanje pristup

riješeno. Stalno nam je nedostajala **Locate** komanda iz dBASE-a koja ovdje nema svog direktnog ekvivalenta, već ste prinuđeni da stalno kreirate nove filtere i da ih po pronađenom zapisu isključujete. U krajnjoj liniji, filter nije ni namijenjen toj ulozi već za selektovanje polja koja će se prikazati na ekranu, pa ako ste na početku rada postavili stalni filter za tu namjenu, dolazite u neizbježnu koliziju. Kada bi bar bilo moguće definirati jedan „stalni" i bar jedan „promjenljivi" filter, rad bi bio mnogo ugodniji.

Komplikovanije manipulacije podacima se mogu vršiti uz pomoć **Query** komandi koje se mogu sačuvati na disku za razliku od filtera. Radi se o najdorađenijoj i najmoćnijoj opciji u *Superbase*-u koja će vam, između ostalog, omogućiti:

- kreiranje kompleksnih filtera koji su često u upotrebi za pretraživanje podataka
- aplikacije koje uključuju više datoteka kreiranjem veza, tako da možete istovremeno imati uvid u više odvojenih baza (tzv. **Multi-file Query**)
- izvještaje o rezultatima pretraživanja i sortiranja kao što su brojenje zapisa, razvrstavanje po raznim ključevima u grupe i podgrupe itd.

Rezultat **Query**-ja se može usmjeriti na ekran, štampač, tekstualnu datoteku ili novu bazu podataka.

Kreiranje upita je u *Superbase*-u pojednostavljeno onoliko koliko je to moguće učiniti. Međutim, da biste ispravno iskoristili sve ono što vam omogućava, biće ipak neophodno proučiti prateću dokumentaciju, ali će zato rezultat biti više nego proporcionalan uloženom trudu.

UVOZ I IZVOZ

Kada i ako pređete iz dBASE-a ili kompatibilnih programa na *Superbase*, nećete se morati odreći starih datoteka, već ćete ih prebaciti u novu okolinu. Uvoz ide bez ikakvih problema. Postoje dva načina za korišćenje DBF datoteka:

- *Superbase* može normalno koristiti ovaj tip datoteka bez ikakve konverzije za sve radnje izuzev za dodavanje novih podataka. Znači, ako ne želite da mijenjate bazični program, možete koristiti *Superbase* samo kao alatku za oblikovanje izuzetno atraktivnih izvještaja.
- S druge strane, omogućena je i konverzija u vlastiti format, pa je rad nadalje moguć kao da su podaci od početka unošeni ovim programom. Probali smo i radi kao što piše.

Podržan je i uvoz tzv. ASCII delimited formata (tj. tekstualnih podataka ograničenih nekim separatorom), *Lotus 1-2-3 WKS* i *WK1*, *Microsoft Excel XLS*. Nabrojali smo samo najvažnija. Takođe, izvoz je izvodljiv u istim formatima.

Budući da je *Superbase 4 Windows* aplikacija, razmjena podataka je omogućena i preko *ClipBoard*-a, ali i posebnog protokola DDE (*Dynamic Data Exchange*) sa programima koji ovaj protokol podržavaju. Sve u svemu, što se tiče veza sa drugim programskim paketima, zime nema.

OBLIKOVANJE MASKE

Modul za kreiranje ulazno izlaznih maski je samostalna aplikacija kojoj se pri instaliranju automatski dodjeljuje posebna sličica, tako da je možete startati iz *Windows*-a, a iz *Superbase*-a jedino preko opcije RUN u CONTROL meniju.

Ako bismo trebali izabrati pravac u kome je *Superbase* otišao najdalje, onda je to nesumnjivo gradnja maski. Omogućeno vam je veoma lako miješanje teksta, grafike u obliku predefinisanih figura te eksternih slika sa diska. Da dnu ekrana je traka sa izborom alata, tzv. TOOLBOX koja nudi masu opcija. Između ostalog tu su i okviri, krugovi, linije različite debljine, izbor različitih uzoraka za površine (**Area**), uključivanje slika, dopisivanja teksta u različitim fontovima i oblicima (podebljano, zakošeno, podvučeno), opcije za uključivanje i pozicioniranje polja iz baze u masku itd.

Jednom postavljene objekte možete po želji pomjerati po ekranu na drugu lokaciju, mijenjati im veličinu ili dimenzije. Naravno, tu su i palete za odabir boja (čak možete praviti nijanse po želji miješanjem tri osnovne boje) i šablona za popunjavanje površi, za oblik i debljina linija itd.

Radi poboljšanja preciznosti na ekranu možete postaviti lenjir (RULER) sa mrežom (GRID) čiju gustinu definišete sami. Umjesto klasičnog krstastog kurzora moguće je dobiti dvije ukrštene linije pod pravim uglom, po ugledu na *AutoCad*.

Vaš izvještaj može sadržavati više različitih strana, npr. na jednoj mogu biti tekstualna polja, a na drugoj samo polje koje uključuju slike. Kreiranu masku možete pogledati i smanjenu, kada se vidi cijela A4 strana, pa ako vam se rezultat sviđa, krenite na štampanje. Čak je dopušteno učitati cijelu sliku da vam služi kao pozadina, a po njoj se onda „lijepe" ostali objekti. Skoro pa naučna fantastika.

Proces kreiranja maski je vrlo jednostavan, da ne može biti lakši. Interesantno je spomenuti da se ista maska može koristiti za štampanje i unos. Po jednoj datoteci sa podacima možete imati neograničen broj maski i koristiti ih zavisno od potreba.

SPECIJALITETI KUĆE

Superbase 4 uključuje mali ali upotrebljiv tekst editor koji vam sigurno neće uspjeti da zamijeni pravi procesor teksta, ali mu to nije ni cilj. Editor se poziva iz UTILITIES menija i radi u posebnom *Windows* prozoru, a uglavnom će biti korišten za sljedeće namjene:

- Za formiranje tekstualne datoteke koja će se priključiti kao eksterna na neko polje. Time se može postići da polje praktički bude duže od maksimalno dopuštenih 4000 karaktera. Kada se u bazi pozove ovo eksterno polje, automatski će se otvoriti tekst editor i učitati datoteka u njega.
- Za pisanje cirkularnih pisama koristeći podatke iz neke baze. Ovo je riješeno veoma zgodno – dovoljno je kreirati nepromjenljivi tekst, a u promjenljivi dio unijeti nazive polja između dva znaka paragrafa. Tu su i sistemne varijable TODAY i NOW za automatsko upisivanje datuma i vremena. Npr. &IME& ili &ULICA& ili &TODAY&.

Mail merge se startuje iz PROCESS menija vodeći računa da i baza podataka i tekst moraju biti otvoreni, i ovdje ćete imati priliku da postavite filter, a omogućen vam je i pregled svih pisama na ekranu prije štampanja.

U tekst editoru, na žalost, ograničeni ste samo na jedan font. Sve se i ne može imati.

Razmjena podataka između dva računara je moguća i serijskom vezom zahvaljujući komunikacionom modulu iz UTILITIES menija. Podaci se prenose modemom ili kablom koji povezuje dva serijska porta. U uputstvu je čak data šema kako se takav kabl može napraviti u kućnom aranžmanu! Autori kažu da će se ovaj drugi način uglavnom koristiti kod računara koji su u istoj sobi.

Bilo kako bilo, u odgovarajućem dijalog boksu ćete podesiti da li računar prima ili šalje podatke, brzinu prenosa (podržane su brzine 300, 1200, 2400, 9600 i 19200 bauda), protokol (XModem, CRC_XM i WXModem) i port (COM1 ili COM2). Modul zna i sam pozvati telefonski broj ako modem tu mogućnost podržava.

Posebno je bitno što dva računara uopšte ne moraju da razmjenjuju samo datoteke *Superbase*-a već i bilo šta drugo. Sve u svemu, ništa revolucionarno, ali se trebalo sjetiti.

O NJEMU ĆE SE JOŠ ČUTI

Program je veoma obiman i u ovom prikazu samo nevelli samo jedan dio njegovih mogućnosti. Ukoliko negdje zapnete, tu je i odličan kontekst-senzitivni help koji će vam pružiti niz informacija. Na žalost, Precision Software nije uključio ništa slično Tutorial-u koji standardno dolazi, recimo, uz *Microsoft Excel*, pa se nadamo da ćemo i to ugledati u sljedećim verzijama. Umjesto njega, možete pogledati vrlo zgodan demo program pod imenom PREVIEW koji se starta opcijom RUN iz DML menija, što će reći da je kompletan urađen u DML jeziku. Reći će vam ponešto o mogućnostima *Superbase*-a, ali i mnogo više o snazi i fleksibilnosti ugrađenog jezika.

Polazeći od pretpostavke da svim korisnicima ne treba jednako moćan program, proizvođač nudi i jednostavniju i jeftiniju varijantu pod imenom *Superbase 2* koji ne uključuje rad u programskom jeziku DML, a i neke druge opcije starijeg brata nedostaju.

Kada smo već kod programskog jezika, o njemu ćemo govoriti drugi puta iz razloga što on dosta zahtijeva poseban napis. Možemo samo reći da se radi o veoma moćnom jeziku sa komandama i sintaksom bliskom bejziku, koji omogućava razvoj aplikacija koje će raditi u *Windows* okruženju.

Red je da na kraju navedemo i par nedostataka. Jedan od njih je nedostatak pregleda strane (**Print Preview**) prije štampanja. Doduše, *Form dizajner* nudi opciju smanjenog pogleda, ali ona nije uključena u glavni program. Trebalo je samo pogledati kako je to izveo Microsoft u *Word*-u i *Excelu*-u. Komande za pretraživanje baze bi mogle biti malo bogatije. Nema UNDO opcije, a ponekad vam tako zatreba. Vrijeme osvježavanja ekrana, zbog pune grafičke okoline, zna biti dosta sporo, naročito ako radite sa mnogo različitih fontova. To je veliki nedostatak svih *Windows* programa, a sve se na kraju svodi na to da morate imati što kapacitetniji računar. I konačno, po nama najveći minus *Superbase* zaslužuje zbog nedostatka (za sada) pravog kompajlera za programe u DML jeziku.

Možemo zaključiti da se radi o dosta moćnom paketu, čak u 1.2 verziji. Još jedna potvrda njegovog kvaliteta jeste činjenica da je izabran za zvaničnu bazu podataka ratne avijacije SAD, a opšte je poznato kakve stroge kriterije vojska namеće po tom pitanju. Dakle, softver o kome će se još čuti.

Korisne adrese

Precision Software Ltd
6 Park Terrace, Worcester Park
Surrey KT4 7JZ, England

Cijena: Direktno kod proizvođača nije poznata, ali smo za vas pronašli najnižu distributersku cijenu od 389 USD kod

Softline International Division
1333 60th Street, Brooklyn, NY 11219, USA

GLEDAM I PROGRAMIRAM

U pretprošlom broju smo izneli prve utiske o novom Borlandovom paketu *ObjectVision*. U međuvremenu, test paketa smo nastavili izradom male aplikacije za jedan praktičan zadatak. Pokazalo se da je *ObjectVision* najjači upravo tamo gde su čuvene softverske alatke najslabije – u izradi takozvanog „korisničkog" interfejsa.

Dejan Dinčić



ObjectVision je namenjen radu pod *Windows 3.0*. Aplikacije se kreiraju crtanjem grafičkih elemenata i korišćenjem menija. Kreirane aplikacije se ne mogu izvršavati nezavisno od samog *ObjectVision* programa. Podržano je više standardnih formata zapisivanja podataka na disk. Osnovu aplikacije čini obrazac (form). Tipične aplikacije pogodne za realizaciju *ObjectVision*-om su evidencije administrativnih obrazaca. Mogu se realizovati veoma složeni obrasci sa dosta zavisnosti među poljima. Paket je snabdeven sa dve veoma dobre knjige: *Tutorial* i *Reference*. *Tutorial* je vrlo efikasan za prvo upoznavanje, dok *Reference* sadrži precizan i formalizovan opis paketa.

CRTANJE MASKE

Zahtev za aplikacijom, koji smo rešavali *ObjectVision*-om, bio je jednostavan: evidencija o uplatama korisnika Sezama – ažurirati datoteku podataka o korisnicima i njihovim uplatama. Podaci o jednom korisniku biće na ekranu predstavljeni jednim obrascem i predstavljaće jedan slog u datoteci. Odlučili smo da kreiramo datoteku u *dBASE* formatu. Da podsetimo, *ObjectVision* podržava i formate *Btrieve*, *Paradox* i ASCII datoteka.

Kako se u *ObjectVision*-u sve vrti oko obrasca (form, maska), počeli smo od crtanja obrasca sa podacima o korisniku. Za taj deo posla koristi se **Form tool**. Poznatim (*Windows*) postupkom izbora opcije u meniju, dodaju se polja u obrazac. Nakon upisivanja naziva polja, na ekranu se pojavljuje krstasti kursor koji vas tera da polje negde postavite. Večina postupaka koji se koriste pri radu sa ovim programom (crtanje linija i pravougaonika, markiranje objekata na ekranu, rad sa menijima) su standardni postupci u *Windows* okruženju i poznati su iz drugih *Windows* aplikacija.

Posao oko crtanja obrasca na ekranu se sastoji iz ucrtavanja polja („kućice" za upis podataka na papirnim obrascima) i crtanja grafičkih elemenata obrasca (razni okviri, linije, rasteri, naslovi, objašnjenja, fusnote). U našem primeru u prazan obrazac prvo smo dodali polje ključa, identifikaciju korisnika Sezama – **username**. Zatim smo, dodajući jedno po jedno, opisali sva polja o korisniku. Da bismo označili šta treba da bude sadržaj svakog polja pri popunjavanju obrasca, uglavnom smo koristili opciju *ObjectVision*-a da nazive polja stavi u njihove okvire. Polja za datum i iznos uplate smo grupisali u tabelu. Potrebna objašnjenja za ova polja upisali smo naknadno, kao običan tekst iznad tabele.

Osim prostog ucrtavanja, neka polja je potrebno i bliže definisati, kako bi se postigla provera sadržaja u trenutku popunjavanja, ili zadovoljio određeni format podatka. Ovo se u *ObjectVision*-u postiže izborom atributa polja u **FieldType** meniju. Za naš mali primer koristili smo neke od (a ima ih više) atributa za numerička i datumska polja. Izbor atributa pokriva uglavnom sve što će vam zatrebati. Pored više formata za prikaz numeričkih, tekstualnih i datumskih polja, tu su i neki tipovi polja karakteristični za **Windows** aplikacije: **SelectionList**, **CheckBox**, **Button**. Ovi tipovi polja su pogodni za podatke čiji je sadržaj jedna od unapred poznatih vrednosti (da/ne, pol i slično).

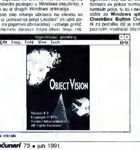
Obrazac koji smo dobili prostim ređanjem polja nije nas baš zadovoljio izgledom. Zato smo pred kraj crtanja ispremeštali polja i stvorili malo kompaktniji vizuelni utisak. Vrlo je korisna mogućnost da se više polja „uhvati" u blok i nad njima izvrši neka od **edit** operacija ili pomeranje celog bloka.

Osim polja za unos podataka, za pravljenja maski mogu se upotrebiti linije, dve vrste pravougaonika, tekst i bitmapirana grafika. Grafika se može napraviti bilo kojim *Windows* programom koji je u stanju da je prenese u zajednički *Windows* **Clipboard**. Za one koji nisu bliski *Windows* koncepciji, da spomenemo da je **Clipboard** poput kopir mašine i razglasne table u isto vreme. Iskopirate parče teksta ili grafika iz jednog programa i okačite na tablu. Zatim u drugom (ili istom) programu skinete sa table šta vam treba i nalepite (**paste**) tamo gde vam treba.

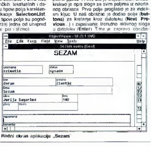
U našem primeru nismo koristili bitmapiranu grafiku, ali smo iskoristili opciju za pisanje teksta i napisali naslov „Sezam" na vrhu obrasca. Naravno, za naslov smo odabrali malo veći font. Za pisanje teksta obrasca u *ObjectVision*-u su vam na raspolaganju svi fontovi koje ste u *Windows* okruženju instalirali. Pomenuti naslov smo još i uokvirili popunjenim pravougaonikom. Tako je obrazac dobio izgled koji smo mogli da prihvatimo za početak.

LOGIČKO STABLO

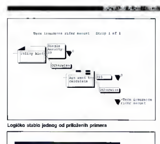
Definisanje datoteke koja će čuvati unete podatke krajnje je jednostavno. Potrebno je izabrati opciju **links**, opredeliti se za jedan od ponuđenih formata, upisati nazive datoteka za podatke, indeks i dozvoliti programu da sam završi posao. A evo šta je *ObjectVision* uradio: kreirao je opis sloga sa svim poljima iz nacrtanog obrasca. Prvo polje proglasio je za indeksni ključ. U naš obrazac je dodao polja (**buttons**) za kretanje kroz datoteku (**Next**, **Previous**...) i zapisivanje trenutno aktivnog sloga u datoteku (**Enter**). Time je zapravo obezbe-



Uvodni ekran



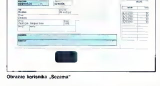
Radni ekran aplikacije „Sezam"



Logičko stablo jednog od priloženih primera



Definisanje veze obrasca sa bazom podataka



Obrazac korisnika „Sezama"

dio sva osnovna funkcije za pamćenje unetih podalaka i lo poljuno bezbolno za korisnika

Drugi način za upravljanja vezom između podalaka iz obrasca i daloteka na disku je korišcenje odgovarajućih funkcija u okviru logike obrasca (**tree**). Funkcije koje odgovaraju odabranom lormalu datoleke mogu se uvrsliti u normalan tok obrade nekog od polja iz obrasca. Sam skup lunkcija za upravljanje datotekama nije previše bogat i svodi se uglavnom na funkcije prelazak na sledeći i prethodni slog, skok na početak i kraj datoteke.

Sledeći korak je bio pokušaj da se definiša logika obrasca. Autor ovih redova, verovatno opterećen programerskom praksom, nije lako lako uspao da prepozna logička struktura koje *ObjectVision* nudi u zamenu za programiranje. A nudi se stablo! Koren jednog ovakvog stabla je neko polje iz obrasca. Uz to polje vezuje se čvor grananja (**branch node**). Na svakoj grani koja rasle iz tog čvora nalazi se po jedan logički uslov. Ako podalak unet u polje zadovoljava uslov, tok izvršenja se prenosi na tu granu. Svaka grana se može račvati i dalje. Na kraju svake grane, kao list, nalazi se zaključni element – (**conclusion**). Ovaj list mora da pruži rezultat šetnja kroz logičko stablo.

Uz polje je moguće vezati i samo zaključak, list, bez prethodnog stabla. Osim izračunavanja, list može sadržati i manipulaciju stringovima, kao i funkcije upravljanja datotekama. Nismo naišli na bilo šta što bi ličilo na iteraciju. U postavljanju uslova grananja i izračunavanjima konste se standardni aritmetički, logički i stringovni operatori, kao i skup ugrađenih funkcija. Ovaj deo kreiranja aplikacije sadrži najviše elemenala koji podsećaju na programiranje.

Većina funkcija koje se ovde koriste je preuzela iz programa *QuattroPro*. To su uglavnom standardne spredšit funkcije. U ovoj implementaciji im je ipak oduzela jedna dimenzija. Dok se u labelama izrazi i funkcije mogu odnositi na neki opseg redova i kolona, u *ObjectVision*-u su argumenti funkcija i izraza samo polja jednog sloga (obrasca), bez mogućnosti da se neka operacija izvede nad više slogova (primeraka istog obrasca) u datoteci.

Interesantna je grafička interpretacija logičkog stabla. Kada se jednom naviknete na logičke strukture *ObjectVision*-a, onda vam grafički prikaz stabla daje vrlo jasnu sliku. Definisane veze su lako uočljive i odnosi među elementima vrlo očigledni. Deo kreiranja stabla se može izvršiti pomeranjem i kopiranjem u okviru grafičke predstava stabla. Grafički uvid u logiku polja moguće je ostvariti i u toku korišćenja gotove aplikacija, pri popunjavanju obrasca, izborom opcije u meniju. To ponekad može pomoći korisniku u odluci kako da unese podatak. Ova mogućnost se može i uskratiti dodavanjem atributa zaštite (**Protection/No Tree Display**) polju.

Nakon ovladavanja osnovama crtanja stabala u *ObjectVision*-u, dodali smo jedan logički list za polje **suma uplata** u naš obrazac. Napisali smo izraz kojim se polju dodeljuje vrednost zbira uplata 1-10. Poljima uplata 1-10 dodali smo stabla sa uslovom da se iznos uplate postavi na nulu ako nije unesen datum uplate.

Za svako polje ucrtano u obrazac moguće je upisati **help** tekst, koji će se aktivirati uobičajenim pritiskom na taster F1. U našoj maloj aplikaciji, **help** tekst smo vezali samo uz polje **status**.

Korisna adresa

MARAND d.o.o.
Kardeljeva ploščad 24, 61000 Ljubljana
Tel. (061) 340-652, 371-114
Fax. (061) 342-757

ŠTAMPANJE OBRASCA

Štampanje definisanog (i popunjenog) obrasca rešeno je u okviru standardnog file menija. Postoji mogućnost da se štampa jedan ili svi obrasci aplikacije. Štampani obrazac izgleda sasvim verno onome što vidite na ekranu i WYSIWIG princip je dosledno izveden. Za štampanje se koriste svi sistemski parametri **Windows** okruženja, kao i fontovi koji su instalirani.

U okviru same aplikacije ne postoje operatori kojima se kontroliše štampa. Štampanje je rešeno samo u okviru sistemskog dela *ObjectVision*-a ispisom obrasca. Ovakav način ispisa obrasca na štampač jeste pojednostavljenje za korisnika. Taj se deo posla, dakle, uopšte ne mora posebno obrađivati. Ipak, to je ograničenje i sužava broj poslova koji se mogu celovito obraditi *ObjectVision*-om na aplikacije tipa popuni obrazac i odštampaj ga. Ono što nedostaje za realizaciju iole složenije aplikacije je mogućnost štampanja proizvoljnog tabelarnog ispisa nad slogovima datoteka sa kojima se inače kroz aplikaciju barata. Tabelarni ispis je osnova većine izveštaja koji se koriste u aplikacijama administrativnog karaktera.

Tabelarni prikaz podataka i inače nije zastupljen u *ObjectVision*-u. Mogu se, naravno, nizovi polja složiti tako da izgledaju kao nekakva konačna tabela, ali tabelarni prikaz u kome svakom redu tabele odgovara jedan slog datoteke ne postoji kao moguća forma kompletnog obrasca. No to je ipak druga priča i druga oblast na koju *ObjectVision* i ne cilja. Ono čemu je namenjen radi dobro.

Sumirajući dosadašnje utiske o *ObjectVision*-u mogli bismo reći da je to alat namenjen ipak jednostavnim i manjim aplikacijama. Jednostavnim aplikacijama u smislu da se ne traže složeni odnosi među datotekama, da nema pretraživanja i odabiranja slogova, i da nije potrebno generisati ni izveštaje takve prirode. Ova ograničenja ne isključuju mogućnost složenih izračunavanja, veza i zavisnosti u okviru jednog obrasca, koje *ObjectVision* efikasno obrađuje.

ObjectVision je alat kojim će korisnik sebi olakšati posao unošenja podataka u neku postojeću bazu. Deo posla koji je obavljan u spredšit programima može se preneti u *ObjectVision*. Moguće je koristi *ObjectVison* aplikaciju za formiranje izvornih dokumenata (npr. tipski ugovori). Može se još i pretpostaviti korišćenje *ObjectVision*-a za kreiranje interaktivnog dela neke obimnije aplikacije, uz upotrebu dodatnog alata kojim bi se realizovao ostatak posla. Programerima, izgleda, još nije odzvonilo.

TRI ARHIVERA

Programi za sažimanje datoteka, popularni arhiveri, predstavljaju jednu od najneophodnijih i najkorišćenijih softverskih alatki. Zato nije čudno što je za PC napisan veoma veliki broj ovakvih programa. Testirali smo najpoznatije među njima i pokušali da odredimo najbolji.

Jovan Bulajić

Arhiviranje je čudesna operacija. Podatke koji se nalaze na disku program za arhiviranje (arhiver, kompresor) može sažeti u datoteku koja je znatno kraća od izvorne. Obrnut proces, dearhiviranje, od arhive nepogrešivo stvara izvornu datoteku. Arhiveri takođe mogu spakovati više datoteka u jednu, a neki čak pakuju i kompletna stabla direktorijuma, pa i ceo hard disk u jednu arhivu (sažetu datoteku). Ovim postupkom se postiže, pre svega, ušteda u prostoru na disku, olakšava se manipulacija datotekama (kopiranje), štede se diskete i vreme za bekap. Modemske komunikacije su praktično nezamislive bez arhivera jer prenos sažetih datoteka traje mnogo kraće nego u njihovom izvornom obliku. S druge strane, sažet oblik nije pogodan za manipulaciju, tj. korišćenje podataka od strane aplikativnih programa (npr. baze podataka, tekst procesora), tako da na disku ipak moramo i dalje držati podatke u njihovom izvornom obliku. Ali, postoje neki standardni formati zapisa grafičkih datoteka (npr. GIF) u sažetom obliku, a dekompresiju vrši interno sam grafički program. Nedavno su se pojavili i programi (o kojima će biti još reči) koji pakuju izvršne (EXE i COM) datoteke tako da se one mogu i startovati (izvršavati) u sažetom obliku jer se same raspakuju u memoriji prilikom startovanja programa.

NA UPOREDNOM TESTU

Razni arhiveri koriste različite metode (algoritme) kompresije podataka (data compression), ali ovom prilikom nećemo ulaziti u razmatranje primenjenih programskih tehnika, već ćemo arhivere posmatrati isključivo sa korisničke strane, tj. sa stanovišta performansi.

Za PC računare je razvijeno mnogo programa za kompresiju, a svaki od njih ima karakterističnu ekstenziju imena sažetih datoteka po kojoj su obično i šire poznati. Pomenućemo najpoznatije: ARC, ARJ, LHA (LZH), PAK, TAR, ZIP, ZOO. Trenutno je na PC tržištu najpoznatiji i najkorišćeniji *PKZip* arhiver (i njegov ZIP format), dok arhiveri popul TAR-a i ZOO-a svoju privlačnost, i pored slabijih performansi, crpe iz kompatibilnosti sa svojim parnjacima na velikim sistemima (VAX-ovima) i UNIX radnim stanicama.

PKZip je naslednik nekad izuzetno popularnog *PKArc*-a i kada se pojavio (1989. godine) doneo je mnoga poboljšanja i novitete u odnosu na prethodnike. Vrlo brzo je postao apsolutni standard na PC-u, i na tom mestu se još uvek drži. Dovoljno je samo da pogledate listu dostupnih datoteka na nekom (bilo kom) BBS-u – ogromna većina programa je ZIP-ovana. Ali nova verzija *PKZip*-a nije objavljena već više od godinu dana, a za to vreme su se već pojavili novi programi koji su ga, po prvim testovima, dostizali i prestizali.

Odlučili smo da razrešimo dilemu (trenutno) najboljeg i uporedno testirali tri arhivera, koja su do sada pokazali najbolje performanse: *PKZip*, LHA (naslednika *LHArc*-a) i ARJ. Na preliminarnim testovima smo isprobali i ostale arhivere, ali su oni ostali daleko iza ova tri. Tačnije, od „odbačenih" samo se PAK približio po performansama *PKZipu*, ali pošto smo tražili odličan program, PAK nije ušao u uži izbor. Nažalost, nismo uspeli da nabavimo Hyper arhiver o kome smo samo čuli, a po nekim testovima koje smo videli takođe se približio *PKZip*-u. Sva tri programa su ili *public domain* (ARJ i LHA) ili *shareware* (*PKZip*), što znači dostupni praktično svakome. Ukoliko imate modem, možete ih skinuti i sa Sezama, a takođe i ostale arhivere i uslužne programa za njih. Verzije programa koje smo testirali su: *PKZip 1.1, LHA 2.12 i ARJ 2.0*.

Kada kažemo performanse, prvenstveno mislimo na stepen kompresije datoteka, dok smo vreme arhiviranja/dearhiviranja stavili u drugi plan. Ovakav stav je vrlo lako obrazložiti na primeru modemskih komunikacija – uzmimo dve različite arhive istog programa koje se razlikuju u dužini za npr. oko 30 Kb (kao u jednom od naših test primera). Dok je razlika u vremenu pakovanja/raspakivanja npr. oko 30 sekundi u korist arhivera koji manje sabija podatke, dotle tih 30 Kb znači dva minuta dužeg prenosa datoteke preko modema (na brzini od 2400 bps, sa ZModem protokolom i dobrom vezom brzina prenosa je oko 15 Kb u minutu). Takođe, ukoliko ste radili sa arhiverima, verovatno ste imali slučaj (koji se, gle beksuza, uvek dogodi baš vama) kada vam je falilo nekoliko Kb da smestite arhivu na praznu disketu – nekoliko dodatnih procenata kompresije (i ušteđenih Kb) uvek dobro dođe.

TRI ARHIVERA...

Od ova tri arhivera, *PKZip* je najstariji. Pojavio se kao naslednik *PKArc*-a, a kao rezultat sudskog spora između SEA i PKWare kompanija oko autorskih prava na ARC format (ekstenziju) datoteka. *PKArc* je svojevremeno bio znatno brži od svog konkurenta SEA Arc-a, ali je sud zabranio firmi PKWare da koristi ARC ekstenziju. Posle podužе pauze, izašao je *PKZip* i još jednom pokazao da od ratova kompanija najviše koristi imaju korisnici. Doneo je radikalno povećanje stepena kompresije, ali je i sporije radio. Poslednja verzija *PKZip*a se pojavila još marta '90, dok su nove verzije LHA i ARJ izašle vrlo skoro: marta odnosno aprila ove godine.

Sva tri testirana arhivera se distribuiraju u okviru samoraspakujućih arhiva. Dovoljno je takvu datoteku iskopirati u željeni direktorijum i startovati je (kao i svaku drugu EXE datoteku). Sama će se raspakovati i napuniti direktorijum svojim sadržajem. LHA je najkraći paket, sadrži samo četiri datoteke (ukupne dužine oko 80 Kb), dok su ZIP i ARJ već solidno popunjeni (oba po više od 300 Kb) dokumentacijom, *utility* programima i drugim korisnim stvarima. U sklopu ARJ a je čak dat i izvorni kod u C-u programa za dearhiviranje. Uz ARJ se takođe dobija i program za konverziju iz formata drugih arhivera (između ostalih tu su i ZIP i LHA) u ARJ format. Zanimljivo je da su kod ARJ i LHA arhiver i dearhiver integrisani u

Još malo o arhiverima

Priča o arhiverima bi teško mogla da prođe bez spominjanja dva srodna programa – SHEZ-a i *PKLite*-a. Prvi olakšava rad i manipulisanje arhivama, dok je drugi kompresor izvršnih (EXE i COM) datoteka. Oba programa su *shareware* i možete ih skinuti i sa Sezama.

SHEZ

Shez je shell program za rad sa arhivama. Najlakše ga je opisati kao neku vrstu *PCTools*-a za arhive. Sve što možete uraditi iz komandne linije (tj. iz DOS-a) pa i više od toga, ovde možete uraditi preko menija, daleko lakše i preglednije. To uključuje pravljenje arhiva, njihov pregled, rad sa pojedinačnim datotekama u arhivama (listanje, editovanje, startovanje) ... i da ne nabrajamo dalje, tu su zaista sve opcije koje bi vam pale na pamet. Podržan je rad sa mišem. Moguće je i automatsko konvertovanje iz jednog formata arhive u drugi. Takođe je i omogućen pregled, tj. kontrola prisustva virusa u arhivama, preko poznatog antivirus programa *Scan* za koji je ugrađena podrška u *Shez*. Poslednja verzija SHEZ-a (6.2, izašla aprila ove godine) podržava sledeće arhivere: *PKZip*, *PKPak* (*PKArc*), *LHArc* (Lha), Zoo, Pak i Arj.

PKLite

PKLite sažima izvršne (EXE i COM) datoteka i znatno ih skraćuje, pri čemu se one mogu i dalje izvršavati, tj. normalno startovati. Izdavač *PKLite*-a je kompanija PKWare, ista ona koja stoji i iza *PKArc*-a i *PKZip*-a. Prosečan stepen kompresije se kreće oko 40%, dakle slično rezultatima ZIP-a za EXE i COM datoteke, s tim da je u *PKLite*-ovane datoteke ugrađen i deo koda za samoraspakivnaje (koje se vrši u memoriji, automatski prilikom startovanja programa). Npr. od primera u našem testu broj 3, *PKLite* je napravio izvršne datoteke ukupne dužine 555,233 bajtova, tj. samo neznatno duže od ZIP a.

PKlite-u možete zadati i da pretraži ceo vaš hard disk i kompresuje sve izvršne programe koje nađe. Ovo može dovesti do znatnih ušteda u memorijskom prostoru, a ukoliko neki programi neće da rade u novoj formi, mogu se nesmetano vratiti u stari oblik (tj. raspakovati). Međutim, ma kako ovo lepo izgledalo na papiru, u praksi se ipak nailazi na dosta problema. Pre svega, veliki broj programa zapisuje parametre konfiguracije, tj. setup na kraj izvršne datoteke, a PKlite nije u stanju to da primati. Pošto programi, naravno, čitaju ili pišu te podatke na hard disk, gde se program nalazi u sažetom obliku, a ne u memoriju gde su raspakovani, obično unište kraj datoteke i program se zaglavi. Zato budite pažljivi prilikom primene *PKLite*-a, a i bez njega ćete sasvim udobno živeti.

Tabela 1 Rezultati uporednog testa

Test	Datoteka (bajtova)	Dužina arhive (bajtova) PKZip 1.1	LHA 2.12	ARJ 2.0	ARJ -m2	Stepen kompresije (%) ZIP	LHA	ARJ	ARJ -m2	Indeks stepena kompresije ZIP	LHA	ARJ	ARJ -m2
1 Ceo disk	20 707 328	10.633.279	10.023 063	9.862 047	10 004 796	48.6	51.6	52.4	51.7	1.00	1.06	1.08	1.06
2 Program	797 418	333 796	288 393	280 841	286 729	61.9	63.8	64.8	64.0	1.00	1.06	1.08	1.08
3 EXE i COM	936 897	553.958	527 602	520.274	526.383	40.9	43.7	44.5	43.6	1.00	1.06	1.08	1.05
4 Tekst	889 739	299.324	292.817	264.909	294 096	66.4	67.1	68.0	68.9	1.00	1.02	1.05	1.02
5 Fontovi	649 179	265 498	226.598	223.883	237 056	66.7	73.3	73.7	72.1	1.00	1.17	1.19	1.12
6 DBF (dBase)	1 108 753	198 696	168.847	156 329	164 401	82.2	84.9	68.7	85.1	1.00	1.18	1.24	1.20

Test	Vreme pakovanja (min.sec) ZIP	LHA	ARJ	ARJ -m2	Vreme raspakivanja (min.sec) ZIP	LHA	ARJ	ARJ -m2	Indeks brzine kompresije+dekompresije ZIP	LHA	ARJ	ARJ -m2
1 Ceo disk	24:24	41 00	38.13	32 30	-	-	-	-	1.00	0.60	0.67	0.75
2. Program	1:00	1.32	1 16	1.05	0 15	0:23	0:25	0 25	1.00	0.63	0.71	0.81
3 EXE i COM	1:09	1 42	1:31	1 22	0 13	0:30	0:32	0:33	1.00	0.62	0.67	0.71
4 Tekst	1 14	1.43	1 23	1 07	0 12	0 22	0:24	0 24	1.00	0.69	0.80	0.95
5. Fontovi	2:07	1 40	1.46	1 10	0:14	0:21	0:24	0:25	1.00	1.17	1.06	1.48
6. DBF (dBase)	2:35	1:54	1:29	1:06	0 16	0:20	0:24	0:24	1.00	1.26	1.51	1.86

okviru jednog programa, dok je kod ZIP-a za dearhiviranje zadužen odvojen program – *Pkunzip*.

Tabela 2

```
X:\IBMPC\ARCHIVER\*.*
---------------------
arj200   exe 186273  ARJ arhiver (v 2.0)
compress zip  18002  Unix kompatibilni arhiver (PC)
dir_arch zip   7986  Daje sadržaj ARC,ZIP,DWC,LBR,LZH,PAK i ZOO fajlova
lha212   exe  43533  LHA (LZH) arhiver (v 2.12)
pklte105 exe  50432  Kompresija EXE i COM datoteka
pkx35a35 exe  72704  PKArc & PKXArc (ARC arhiver)
pkz110   exe 149219  PKZip arhiver (v 1.10)
shez62   zip 141481  Shell za rad sa zip, arc, arj i ostalim arhiverima
zoo201   exe  69248  VAX kompatibilni arhiver (IBM PC)
```

Arhiveri koje možete naći na Sezamu.

... NJIHOVE MOGUĆNOSTI ...

Svi arhiveri imaju manje-više slične osnovne opcije: dodavanje datoteka u arhivu, njihovo brisanje iz nje, listanje sadržaja, dodavanje komentara, provera ispravnosti (integriteta) datoteke... Sva tri poseduju i napredne opcije poput šifrovanja prilikom arhiviranja, tako da datoteku može otpakovati i podacima pristupiti samo onaj kome je poznata šifra. Takođe, moguće je spakovati kompletna stabla direktorijuma, pa i čitave hard diskove, u jednu arhivu. Ovo je vrlo zgodna opcija za pravljenje bekapa.

Moguće je kreirati i samoraspakujuću (*self extracting*) arhivu. Tako napravljena datoteka ima ekstenziju .EXE i startuje se kao i svaki drugi program, ali tom prilikom automatski se raspakuje i kreira izvorne datoteke. Ovakva arhiva je samo nekoliko kilobajta duža od obične (ne-samoraspakujuće), pa se mogu tako spakovani programi distribuirati u *self extracting* (SFX) obliku bez potrebe za dearhiverom.

ARJ i ZIP omogućuju i popravak, tj. spašavanje onoga što se spasti može iz oštećenih arhiva, dok LHA nema ovu, vrlo značajnu, opciju. Do oštećenja može doći prilikom havarije diska ili (vrlo čest slučaj) greška prilikom prenosa modemom. Naime, loša veza i loš protokol za prenos mogu dovesti do razlika između originalne arhive i one koja je preneta na vaš računar preko modema, a takođe može i usled pucanja veze doći do prenosa samo dela arhive. Tada dearhiveri ne mogu obaviti svoj posao, pa treba pokušati rekonstrukciju datoteke. Rezultati su obično dobri: i pored grešaka uspeda se spase većina podataka. Kod tekstova se čak skoro ništa i ne izgubi (sem, naravno, dela koji nije ni prenet), jer se obično ubaci samo malo đubreta (nevezanih karaktera), što se lako može rekonstruisati.

ARJ i ZIP poseduju i više različitih načina kompresije koje možete birati. Razlika se uglavnom svodi na brži rad na račun slabijeg stepena kompresije. *PKZip* poseduje dva ovakva metoda, dok ih ARJ ima čak četiri. Prilikom testiranja smo kod ARJ-a uzeli u obzir dva od tih načina koji daju najviši stepen kompresije, pri čemu drugi metod (ARJ -m2) ima nešto slabiji stepen kompresije od osnovnog (ARJ -m1), ali je zato dosta brži. U dokumentaciji ARJ-a se drugi metod spominje kao kompromis (i to dosta dobar) između brzine i kompresije.

Samo kod ARJ-a smo naišli na jednu vrlo značajnu opciju. Naime, moguće je definisati maksimalnu dužinu dela arhive koju želimo napraviti i onda će sam ARJ omogućiti njenu podelu na delove i kasnije ponovno sastavljanje. Tako možete zadati maksimalnu dužinu na npr. 1200 Kb (tj. 1.2 Mb, kapacitet HD diskete) i raditi bekap velike količine podataka na diskete. Svaki put kada deo arhive dostigne određenu dužinu (1.2 Mb), ARJ će od vas zatražiti da ubacite novu disketu i nastaviće sa postupkom. Dobijeni delovi imaju ekstenzije .ARJ, .A01, .A02... i prilikom dekompresije samo treba opet stavljati diskete istim redom. Ova opcija se zove *enable multiple volumes* i dobija se prekidačem **-vxxxx**, gde je **xxxx** maksimalna dozvoljena dužina datoteke u Kb. Tako izbegavate korišćenje programa za cepanje datoteka prilikom kopiranja velikih arhiva, a ni na disku ne morate imati dovoljno prostora za privremeno smeštanje takve arhive – moguće je uraditi bekap potpuno popunjenog hard diska.

... I PERFORMANSE

Testirali smo efikasnost arhivera na različitim tipovima datoteka. Naglasak smo stavili na stepen kompresije, ali smo takođe merili i brzinu rada. Dobijene rezultate možete videti u tabeli 1. Svi testovi su izvršeni na 386SX / 16 MHz računaru, sa hard diskom vremena direktnog pristupa 23 ms (Core Performance Index = 6.3), naravno bez instaliranih ikakvih ubrzivača diskova (cache programa). Opredelili smo se za testiranje na hard a ne RAM disku (kako se ponekad radi) jer nam je cilj bio da ispitamo ponašanje u uslovima u kakvim se arhiveri obično koriste, a to je vrlo retko RAM disk.

Stepen kompresije je izražen kao procenat ZA koliko su stišnjene izvorne datoteke, dok neki više vole oblik NA koliko su spakovane (takav oblik ispisuju ARJ i LHA). Npr. ako je dužina datoteka 100 Kb a dužina arhive 30 Kb, one su spakovane ZA 70%, tj. NA 30%. Sasvim je svejedno u kom obliku navodite rezultate (naravno, taj oblik treba naglasiti), naša orijentacija na „ZA %" je samo rezultat navike još od dana *PKArc*-a do *PKZip*-a.

Prvi od testova koje smo izvršili bilo je arhiviranje cele jedne particije hard diska, sa oko 20 Mb podataka. Ne možemo garantovati da je odnos raznih tipova datoteka (programa, tekstova, podataka) bio sličan onom na vašem disku, ali smo ovo uzeli kao prosečan slučaj hard diska kome je potrebno napraviti rezervnu kopiju. Arhivirali smo kompletno stablo direktorijuma hard diska, ali samo u ovom testu nismo merili i vreme raspakivanja, pošto na drugim particijama nismo nikako mogli da nađemo slobodnih drugih 20 Mb, a nije nam se brisalo ništa od podataka. Stepeni kompresije su se kretali oko 50%, što je već solidan rezultat – u ovom slučaju potrebno vam je 8 do 9 disketa za bekap umesto 17. ARJ je pokazao najbolji stepen kompresije (8% bolji od ZIP-a), dok je *PKZip* ispao dosta brži od ostalih. LHA nije mnogo zaostao za ARJ po procentu kompresije (6% bolji od ZIP) ali je zato ispao pravi sporać po brzini – čak 68% sporiji od ZIP a što je zaista prevelika razlika, bez obzira što smo efekat brzine stavili u drugi plan. Zanimljivo je da je apsolutni brzinski šampion ovde bio brzi metod kompresije *PKZip*-a (**-es**, **shrink only**) koga nismo uzeli u obzir kod uporednog pregleda. On je ceo posao završio za samo 9 minuta i 9 sekundi – ovo je čak 2.5 puta brže od default metoda *PKZip*a, ali je zato i kompresija 25% lošija (arhiva je bila dugačka 13 290 322 bajtova).

Za drugi test smo uzeli nešto što bi moglo predstavljati prosečan program koji se distribuira u arhiviranom obliku – izabrali smo komunikacioni program *Telix* (verzija 3.12). I ovde je ARJ ispao najbolji, a ZIP najbrži (naročito po vremenu dearhiviranja). Indeksi su otprilike ►

na nivou prvog testa, jedino je razlika u brzini nešto manje izražena. Ovde se stepen kompresije kretao nešto iznad 60%, pa vi sada izračunajte koliki su to dobici prilikom prenosa modemom.

Za treći test smo pokupili nekoliko raznih EXE i COM (izvršnih) datoteka. Za taj tip datoteka je karakterističan niži stepen kompresije, mi smo dobili rezultate oko 40%. Rezultate pogledajte sami, nema potrebe da ih mnogo komentarišemo.

Četvrti test smo uradili sa raznim tekstualnim datotekama. Ni ovde nije bilo nekog pravila u izboru, pokupili smo nekoliko dokumenata (u čistom ascii obliku), uputstva za testirane arhivere, izvode iz konferencija na Sezamu. Stepen kompresije je ovde već bio preko 65%, a kod ovog testa su zabeležene i najmanje razlike između merenih rezultata za sva tri arhivera.

Još jedna grupa datoteka karakteristične strukture su fontovi, koje smo uzeli za peti test. Konkretno, uzeli smo sve varijacije helvetike iz *Venture*. Rezultati su za nas bili vrlo iznenađujući – ZIP je ispao ne samo lošiji nego čak sporiji i od LHA i od obe testirane metode ARJ-a. Čak šta više, dotadašnji šampion sporosti LHA se pokazao brži i od ZIP i od ARJ-a.

I za poslednji test smo uzeli nekoliko DBF (dBASE) datoteka pokupljenih sa raznih strana. Karakteristično za DBF datoteke je velika neekonomičnost, tj. pravo rasipanje nepotrebnog prostora, zbog čega se one mogu kompresovati u veoma velikom procentu. U našem slučaju, taj procenat se kretao oko 85%. Na ovom testu je *PKZip* potpuno podbacio – bio je ubedljivo najlošiji i najsporiji. ARJ je datoteke spakovao 24% bolje uz istovremeno 51% veću brzinu rada u odnosu na ZIP. Drugi metod ARJ-a se pokazao bržim čak 86% uz 20% bolju kompresiju, dok je i LHA bio znatno bolji od ZIP-a. Ako su nas rezultati prethodnog testa iznenadili, ovde smo već bili potpuno zbunjeni. *PKZip* se ponašao mimo svake logike – em je radio lošije, em je i to loše radio strašno sporo. Iz ovih poslednjih rezultata smo izveli prilično jasan zaključak – ZIP se slabo snalazi sa datotekama koje se mogu sabiti u velikom procentu.

KO JE NAJBOLJI

Posle nekoliko stotina megabajta pakovanja i raspakivanja podataka (izvršeno je znatno više testova nego što je to prikazano u tabeli) čini nam se da možemo izraziti prilično konkretan stav u pogledu trenutno najboljeg arhivera za PC.

Što se tiče mogućnosti tj. opcija, u osnovnim stvarima su uglavnom sva tri arhivera iste. Mana LHA je nemogućnost spašavanja oštećenih arhiva, kao i postojanje samo jednog metoda kompresije. Plus za ARJ je mogućnost podele arhive na više delova željene dužine.

Po svim dobijenim rezultatima, ARJ je pokazao najbolji stepen kompresije, ZIP još uvek uglavnom najbrži, mada su tu brzinu pomutili rezultati poslednja dva testa. LHA je pokazao dosta dobar stepen kompresije, uvek bolji od ZIP-a i blizu ARJ, ali se zato pokazao i spor. Drugi metod kompresije ARJ (ARJ -m2) se pokazao kao odličan kompromis između želje za većim stepenom pakovanja i brzine rada.

Bez sumnje, ARJ je trenutno najbolji izbor za arhiver koji ćete uglavnom koristiti. To ne znači da će tako istisnuti ZIP, koji će verovatno još neko vreme ostati standardan oblik PC arhive. Ipak, novog ZIP-a nema već više od godinu dana i logično je očekivati (ili bar nadati se) da će doći do nekih njegovih poboljšanja. Od još jedne trke za performansama, najviše koristi ćete opet imati vi.

PRIVATNI STOMATOLOG

(autori: Vladimir Krstonošić i Pavle Peković)

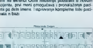
Domaća softverska produkcija se ne svodi samo na računovodstvo i knjigovodstvo – pred nama je ovoga puta program namenjen privatnim stomatološkim ordinacijama.

Program *Privatni stomatolog* se isporučuje na jednoj XT disketi praćenoj atraktivnim uputstvom, koje je po obimu malo (20 A5 strana), ali po sadržaju sasvim dovoljno za uspešno korišćenje programa. Instalacija se svodi na umetanje diskete i kucanje komande A:INSTALL X (nemojte da se prevarite pa da otkucate INSTALL X, jer će instalacija lepo proći ali će sve ostati neinstalirano!), gde je X oznaka diska na koji se program prenosi. Trenutak docnije sam program i potrebne baze podataka naći će se u direktorijumu X:\PZ i u njegovom poddirektorijumu X:\PZ\FILES. Program se docnije startuje komandom PS, a dalja „navigacija" je sasvim zahvaljujući jednostavnom sistemu menija zaista pristupačna svakome.

Privatni stomatolog radi kako na *Hercules*-u tako i na EGA/VGA kartama pri čemu se na VGA efektno koriste boje, a na *Hercules*-u je prirodno ceo svet crno-beo ili barem „monohroman", što sa funkcionalne strane niukoliko ne umanjuje mogućnosti programa. *Privatni stomatolog* nije zaštićen od kopiranja, ali se u okviru osnovnog ekrana stalno pokazuje ime stomatološke ordinacije, a osim toga, u krupnom planu, i imena autora. Prvo je sasvim prirodno, drugo malo pretenciozno – trebalo je, ako ništa drugo, da ime ordinacije bude u centru, a imena autora u dnu ekrana!

Rad sa programom svodi se na izbor jedne od stavki menija **Pacijent**, **Zakazivanje** i **Račun**, dok je za napuštanje programa zadužena opcija **Izlaz**.

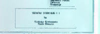
Meni **Pacijent** omogućava unošenje podataka o novom pacijentu – unosi se prezime, ime, adresa, mesto, telefon, datum rođenja, pol a predviđeno je i nekoliko redova za proizvoljne napomene. U okviru svakog od polja mogu se unositi YU slova, ali ona ne figurišu ni u jednoj masci. Obzirom da program ne omogućava nikakvo sortiranje slogova, sasvim je nebitno u skladu sa kojim standardom su YU slova raspoređena na vašem sistemu – važno je samo da ih vidite na ekranu! Osim unošenja podataka o novom pacijentu, prvi meni omogućava i pronalaženje pacijenta po delu imena i ispisivanja kompletne liste pacijenata u bazi.



Izaberi Zub

Kada izaberete pacijenta, pojavljuje se novi meni u okviru koga možete da vidite podatke o njegovim zubima, opšte podatke, istoriju bolesti, zakažete sledeću intervenciju ili ispišete račun. Najzanimljivija opcija je prva – **Podaci o zubima**. Umesto neke klasične ASCII maske, *Privatni stomatolog* po aktiviranju ove opcije prelazi u grafički mod i iscrtava veoma efektnu šemu stanja pacijentovih zuba u okviru koje su obeležene bolesne površi, karijesi, krunice, proteze i sve ono što se nekada upisivalo u svakome dobro poznate žute kartone. Zapravo, autori su uspeli da „snime" dizajn kartona, tako da će se svaki stomatolog koji ga koristi naći u poznatom ambijentu. Samo unošenje podataka se svodi na šetanje kurzora i izbor stavki iz menija što je pomalo neprijatno kada treba uneti podatke o većem broju zuba, ali docnije obeležavanje intervencija se svodi na par pritisaka na kurzorske tastere.

Štampanje računa je sasvim jednostavno – unese se šifra usluge, iznos i datum (ako je različit od današnjeg), uključi štampač i... sve je tu! Ordinacijama je ostavljeno da samostalno definišu listu usluga i cena, a zatim te tekstove ubacuju u račun prostim pritiskom na F4. Što se tiče izgleda odštampanog računa, nemamo primedbi premda moramo da kažemo da smo imali muke da ga do njega dođemo – program ne omogućava konfigurisanje štampača kao LPT2 pa smo u nedostatku programa koji „razmenjuje" LPT1 i



"LPT2" smatrali najjednostavnijim da isključimo računar i zamenimo kablove!

Meni **Zakazivanje** omogućava pregled radnog vremena za proizvoljni dan i unošenje podataka o zakazanoj intervenciji (ime pacijenta se automatski upisuje iz baze, treba samo zadati vreme i eventualni komentar) kao i pregled zakazanih intervencija za zadatog pacijenta. Svaki termin se po potrebi može promeniti ili izbrisati.

Meni **Račun**, osim unošenja cenovnika i štampanja računa, omogućava i pregled nenaplaćenih potraživanja, kao i pregled po pacijentu odnosno utvrđivanje da li je neko uredan ili neuredan platiša.

Dokumentacija je, kako rekosmo, sasvim dovoljna za komfornu upotrebu programa i koncipirana je gotovo idealno – najpre su objašnjene sve komande programa, a zatim opisano šta treba raditi u trenutku kada dolazi novi pacijent, kada se zakazuje intervencija, štampa račun itd. Uostalom, program je toliko jednostavan za upotrebu da će dokumentacija biti potrebna samo u prvim danima navikavanja.

Sve u svemu, program *Privatni stomatolog* je dobrodošlo osveženje na domaćem softverskom tržištu. Iz mnoštva ga izdvaja pre svega izvanredno rešen rad u grafičkom modu, dobro sagledavanje potreba jedne stomatološke ordinacije i relativno jednostavne upotreba. Moguća poboljšanja bi se mogla očekivati, pre svega, kroz jednostavnije unošenje podataka o većem broju zuba (možda da se podrži i miš?), a zatim i kroz neke opcije za sumarni pregled i analizu poslovanja u pojedinim mesecima ili drugačije zadatim intervalima. Ali sve je to neka vrsta nadgradnje – program *Privatni stomatolog* preporučujemo svim stomatološkim ordinacijama koje razmišljaju o kompjuterizaciji svoga poslovanja.

Kontakt sa autorima

Sa autorima programa možete kontaktirati na telefone (011)530 914 i (011)863 933 ili preko Sezama uputite poruku za *vkrstonosic* ili *ppekovic*.

Dejan Ristanović

DOBRO DOŠLI U VERMONT

Uz gotove biblioteke C potprograma – koje obuhvataju sve što vam srce može poželeti: grafiku, komunikacije, baze podataka, matematičke funkcije – danas nije teško biti programer. Svaka aplikacija, doduše, ima svoje specifičnosti i drugačiji korisnički interfejs, a *Vermont Views Designer v2.0* ja namenjen upravo za takva poslove.

Siniša Ćurčin

Firma Vermont Creativa Softwara ja počela da razvija editor za kreiranje menija i maski za unos podataka 1987. godine. Biblioteka sa preko 500 funkcija je završena 1989 godine, godinu dana kasnije i *Vermont Views Designer*, a na i kraju *Vermont Views GraphEx* namenjen za graficke aplikacija.

Koje su osnovne prednosti ovog programa? Jednostavno kreiranje i dizajniranje menija, maski za unos podataka i slično, mogućnost lakog i brzog modifikovanja programa, generisanje izvornog koda u C-u za kreirani meni ili masku, kompatibilnost sa Microfosl (C v5.0, v5.1 i C v6.0) i Borland (*Turbo C v2.0 – Turbo C++ v1.0*) C kompajlerima. Pored verzije za PC-DOS postoje i verzije za UNIX, VMS i XENIX.

Program se isporučuje na četiri diskete 5.25" 1.2Mb, a instalacija se vrši pomoću instalacionog programa. Da bi ste mogli da instalirate dizajner potrebno je da imate neki od navedenih kompajlera i najmanje 5 megabajta slobodnog prostora na tvrdom disku. Cena *Vermont Views Designer*-a je 599$, a *Vermont Views GraphEx*-a 149$.

Prvi objekat koji se definiše prilikom dizajniranja je forma. Forma predstavlja ili meni ili masku za unos podataka. Forma se sastoji od polja. Za svaku formu se mogu definisati korisnička funkcije koje će se pozivati za vreme rada u formi (na početku rada sa formom, za vreme rada i na kraju rada). Maska za unos podataka može biti veca ili ista kao prozor na ekranu u kojem sa prikazuje. Ako je maska veća, onda se ona naziva virtuelna maska i kretanjem između polja menja se vidljivi deo maske. Polje određuju dva parametra: tip i format. Ako polje sadrži više redova, onda se naziva memo. Tip polja može biti: logički, celobrojni, realni, datum, vreme..., a format se može definisati prema tipu. Na primer, format za unos telefonskog broja može biti

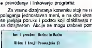
(UUU)UUU-UUUU

Postoji mogućnost da se prikaže odgovarajuća poruka kada se kurzor nađe na polju, da se definiše lista vrednosti, boja polja i funkcije koje će se izvršavati za vreme dok se kurzor nalazi na polju (slično kao i za masku). Kreiranje menija sada zaista ne predstavlja nikakav problem, jednostavno se polja rasporede na formi (polja predstavljaju opcije menija), za svako polje se definiše odgovarajuća akcija (može se pozivati maska za unos podataka ili novi meni) i posao je završen. Jednom urađena forma se može kasnije modifikovati (promena veličine, mesta na kojem se nalazi, karakteristika polja, itd).

Kreiranje aplikacija se obavlja u nekoliko koraka:

- dizajniranje formi (menija i maski)
- generisanje koda
- kreiranje korisničkih funkcija
- prevođenje i linkovanje programa

Za vreme dizajniranja korisniku stoji na raspolaganju jednostavan meni, a na dnu ekrana se javljaju poruke i podaci koji olakšavaju rad sa dizajnerom. Akcije se mogu izabrati pomoću menija ili kombinacijom tastera. Može se dizajnirati više formi istovremeno, što je naročito korisno prilikom kopiranja objekata sa jedne na drugu formu, određivanja relativnog položaja formi, kreiranja sistema menija sa podmenijima. Sve forme se smeštaju u biblioteke. Primer forme za unos podataka je prikazan na slici 2. Nakon završenog kreiranja forme pomoću programa VVDRPT.EXE se može generisati dokumentacija (report) za biblioteke ili forme. Sledeći korak je generisanje koda korišćenjem *VVDCODE.EXE*. Rezultat je C program RAC.C i heder PODACI.H.



Slika 2 Primer forma za unos podataka

Dobijeni izvorni kod se sada može jednostavno modifikovati prema želji i, naravno, potrebama i povezivati se sopstvenim C aplikacijama. Autor ovog članka je bez ikakvih problema na ovaj način generisao program za unos podataka u bazu *ORACLE*. Prevođenje se vrši sa sledećim opcijama:

- *Turbo C v2.0*:
  tcc -ml -I.;c:\tc\include;c:\views2\include – Lc:\tc\lib;c:\views2\lib ime_dat
- *Microsoft C v5.1*
  cl /AL /W3 ime_dat.c vvdevl.lib llibce.lib vvdevl.lib

Vermont Views GraphEx je dodatna biblioteka za DOS verziju *Vermont Views v2.0*. Dodatne funkcije omogućuju prikazivanje grafika na ekranu i korišćenje grafičkih biblioteka *Essential Graphics*, *South Mountain*, *Halo*, *Media Cybernetic*, *MetaWINDOW Metagraphics Software* ali i ostalih biblioteka uz izvesne prepravke.

Kontakt adresa

Vermont Creative Software
Pinnacle Meadows Richford, Vermont 05476
Tel. 802-848-7731 Fax 802-848-3502



Slika 1 Jednostavan meni za oblikovanje formi – menija i maski za unos podataka



Slika 3 Primer aplikacije u programu *Vermont Views GraphEx*

Nastavlja se na strani 67

OBJEKAT I KAKO GA PREDSTAVITI

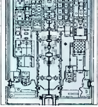
Ako igde ne važi IBM-ova krilatica da je računar najgenijalniji produkt ljudake lenjosti, onda je to CAD tehnologija. Možda bi umesto lenjosti bolje zvučalo nemoći, jer jednostavno čovek nije u mogućnosti da, baz obzira na svoje želje, reši problem strujnog polja oko vazduhoplova, simulira proboj metka kroz metalnu ploču... U ovom članku biće više reči o predstavljanju objekata u CAD-u, gde se najbolje može potvrditi prethodna teza.

Slavko Korać i Milan Bojanović

Kao sto je u prethodnom članku ove serije već naglašeno, dizajniranje odnosno modeliranje podrazumeva generisanje i procesiranje određene klase informacija. O kojim je informacijama ovde reč? Pre svega, informacije se odnose kako na realne fizičke objekte tako i na funkcionalne sisteme i opisuju njihovu formu i (ili) funkciju. Bazirani na ovim informacijama, slede procesi specifikacije, kreiranja, analize i sinteza određenog modela, to jest konstruktivnog rešenja.

FORMA I FUNKCIJA

Pošto smo pomenuli pojmove forme i funkcije, treba naglasiti da oni predstavljaju osnovne karakteristike nekog objekta koje je praktično nemoguće posmatrati odvojeno. Kojoj od ovih karakteristika će se u procesu projektovanja dati veća težinska vrednost zavisi od same oblasti (mašinstvo, elektronika, građevinarstvo, arhitektura...) kao i od individualnog pristupa samog projektanta. Međutim, kako je za potrebe pojašnjavanja pojmova ipak neophodno napraviti izvesnu razliku između ove dve karakteristike, treba naglasiti da će u daljem tekstu gro pažnje biti posvećen predstavljanju forme objekta iz jednostavnog razlo-



Slika 2 Zabranjeni grad u Pekingu

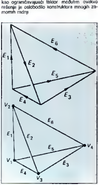
ga što je na ovom polju u poslednje vreme zabeležen značajan napredak.

U pokušaju da se prikaže forma određenog objekta odnosno sistema, projektant se neizbežno susreće sa problemom definisanja i prepoznavanja, a isto tako i sa problemom formiranja dovoljno vernog modela željenog objekta. Sa razvojem CAD sistema ovi proble-



Slika 1 Homologne forme – različiti tipovi automobila koji izvršavaju istu funkciju

mi su postajali sve akutniji, tako da je rešenje traženo u restrikciji broja mogućih formi osnovnih objekata, pomoću kojih se generišu složeniji. Rezultat ovoga je da se, na primer, pomoću sfere, konusa, valjka, kvadra i sličnih geometrijskih primitiva mogu kreirati veoma složeni modeli. Na prvi pogled ovo može delovati kao ograničavajući faktor, međutim, ovakvo rešenje je oslobodilo konstruktora mnogih zamornih radnji.



Slika 3 a) deklarativna forma piramide, b) proceduralna forma piramide

OBJEKTI I ATRIBUTI...

Fizički objekti koji se modeliraju razlikuju se kako po svojoj veličini (od mikročipova pa do čitavih gradova), tako i po svojoj kompleksnosti (zavrtanj i mreža super-kompjutera). Unutar ovih granica se nalaze motorna vozila, mostovi, brodovi, avioni, mašine alatke, roboti, kao i čitav skup najrazličitijih potrošačkih proizvoda. Pored fizičkih, postoji i čitava klasa takozvanih „nematerijalnih" objekata, u koju spadaju, na primer, komunikacione mreže i sistemi za kontrolu letenja, zatim organizaciono-strukturne mreže multinacionalnih kompanija, ali i kompjuterski programi i softverski sistemi.

Sa napretkom tehnike i tehnologije, praktično svi modeli podležu stalnim promenama svoje forme, tako da je besmisleno govoriti o nekoj konačnoj ili „fiksnoj" formi objekta. Neverovatnu brzinu ovih promena čitalac najlakše može uočiti ako se samo osvrne na žensku modu, uz napomenu da i u ovoj oblasti postoje prilično razvijeni softverski paketi kojima se menjaju dezeni materijala, krojevi i slično.

U vezi forme, ali i funkcije objekta, treba naglasiti da se često događa da objekti, uzajamno veoma različiti po formi, vrše praktično istu funkciju, kao i obrnuto, da objekti izuzetno slične forme imaju potpuno različite funkcije.



Slika 5 a) Žičani model, b) površinski model, c) čvrsti model

Na ovaj način smo došli do pojmova homolognih i analognih formi.

Primer za homologne forme dat je na slici 1, gde su prikazana dva različita modela automobila (objekti koje karakteriše ista funkcija a različit spoljašnji izgled). Na slici 2 prikazana je jedna analogna forma. Radi se o shemi zabranjenog grada u Pekingu (slika 2), koja neodoljivo podseća na formu modernih integrisanih kola, na primer Inmos Transputer-a, iako se suštinski radi o objektima za koje je teško reći da imaju bilo šta zajedničko. Iako nemamo više nameru da čitaoce *Računara* zamaramo filozofskom komponentom pojmova homolognih i analognih formi, ipak treba napomenuti da homologene forme karakterišu parametrizova-



Slika 4 Senčeni model automobila PEUGEOT 405

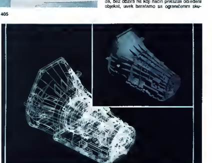
ne oblike objekata (o ovome će biti reči kasnije), dok su analogne nezamenljive za potrebe simulacije

...I NEŠTO VIŠE O NJIMA

Jedan od osnovnih problema koji se javlje u kontekstu CAD-a tiče se odnosa između objekta i njegovih atributa, kao što su oblik, boje, tekstura itd. Od mnogobrojnih aspekata u kojima se ovaj problem može javiti navešćemo četiri osnovna

- objekat je u suštini definisan svojim atributima
- atribut realno postoji samo u odnosu na objekat
- atribut je suštinski definisan pomoću objekta koji karakteriše
- objekat (ili atribut) može biti prikazan pomoću više različitih formi.

Prvi aspekt sugeriše da se objekat može identifikovati pomoću skupa atributa koje poseduje. Međutim neosporno je da realni objekat postoji bez obzira na svoje atributa. Problem leži u činjenici da je praktično nemoguće uzeti u obzir sve atribute koji karakterišu neki objekat, što znači i da je nemoguće dati njegov apsolutno veran prikaz. Drugi aspekt se tiče činjenice da atributi kao što su „okrupao", „četrvrtast", „čvrst", „topao", itd. ne predstavljaju delove nekog objekta, iako ga opisuju. Ovi atributi mogu da postoje samo u relaciji sa objektom. Treći aspekt sugeriše da atribut može biti identifikovan pomoću skupa objekata koji taj atribut poseduju. Ovaj skup objekata se kontinualno menja, i intuitivno je jasno da je neograničen iako se za praktične potrebe mora usvojiti da je konačan. I na kraju, četvrti aspekt se bavi pitanjem da li postoji neki specijalni (kanonični) način za prikazivanje svakog objekta (i svakog atributa) iz kojih bi bilo moguće izvesti sve ostale. Ako ne postoji, na koji način su različite predstave nekog objekta međusobno povezane? Ovo će biti jasnije ako se setimo da baza kružnog cilindra ima eliptičan oblik u različitim dvodimenzionalnim projekcijama, ali u isto vreme poseduje jedinstven kanonični atribut „kružne" u određenoj specijalnoj projekciji, iz koje sve ostale mogu da budu izvedene



Slika 5. Transformacija prelaska žičanog modela u čvrsto telo

> **NAŠA GREŠKA**
>
> Tehničkom omaškom u prvom nastavku serije o CAD/CAM projektovanju, koji smo objavili u prošlom broju pod imenom *CAD, CAM, CAS, CAE, CIM*, izostala su imena njenih autora, Slavke Korać i Milana Bojanovića. Redakcija se izvinjava autorima i čitaocima.

Iako u svemu ovome možda ima više filozofije nego što bi trebalo, ipak je bilo neophodno osvrnuti se na pojmove objekta i njegovih atributa, jer oni predstavljaju bazu za dalju diskusiju o modeliranju pomoću računara, odnosno CAD/CAM tehnologiji.

PRIKAZIVANJE I MODELIRANJE

Da bi uopšte bilo moguće pristupiti procesu projektovanja (modeliranja), analize, sinteze i simulacije, neophodno je na neki način identifikovati i predstaviti delove oblikovanog objekta, njihove osobine, ponašanje i međusobne odnose. U određenom smislu, modeli realnih objekata predstavljaju njihove analogne opise koji u kontekstu CAD-a moraju biti veoma precizno formulisani

Kako je svakodnevni jezik previše opširan i neprecizan da bi obezbedio odgovarajuću platformu za opis određenog modela, posve je prirodno da se, zbog preciznosti koji pružaju, prihvate matematički koncepti i tehnike za ovu svrhu. Međutim, ni u kom slučaju ne treba zanemariti činjenicu da jezik matematike poseduje određena ograničenja. Prema tome matematički opis nekog objekta ili sistema je uvek ograničen i nikako ga ne treba brkati sa stvarnim, realnim objektom ili sistemom. Ovo znači da, bez obzira na koji način prikazali određeni objekat, uvek baratamo sa ograničenim skupom njegovih atributa

Imajući sve unapred rečeno na umu, sasvim je moguće kreirati veštački (ograničen) skup matematički generisanih objekata, i u tom smislu dobiti kompletnu definiciju takvih objekata. Ako se sa ovakvog modela ikada pređe na fazu konstruisanja realnog objekta, mora se prihvatiti činjenica da će takav objekat imati čitav niz atributa i osobina koje pomoću pomenutog matematičkog modela nisu mogle ni približno biti predviđene. Kao dobar primer može se navesti slučaj modeliranja elektronskog kola, formiranog u skladu sa preciznim geometrij- ▶

skim pravilima koje ne funkcioniše zbog pojave toplotnih probleme (atributi koji nisu uzeti u razmatranje)

Kako svi realni objekti postoje u trodimenzionalnom prostoru geometrijsko prikazivanje je od primarnog značaja čak i kada su u pitanju negeometrijski aspekti ili ponašanje.

DEKLARATIVNO I PROCEDURALNO

Postoje dva fundamentalno različita načina na koje određeni objekat ili sistem može biti prikazan. Radi se o takozvanom deklarativnom i proceduralnom pristupu. Već iz samog naziva se može zaključiti da deklarativni način podrazumeva opisivanje konačne forme objekta, dok proceduralni pristup treba shvatiti kao opis postupaka, to jest procedura neophodnih za formiranje željenog objekta. Jedan primer iz svakodnevnog života: izgled gotove torte i recept za njeno pripremanje, ili gotov džemper i mustra za njegovo pletenje. Analogija primerena CAD tehnologiji bi bila između objekta prikazanog pomoću njegovog geometrijskog oblika (tačke, linije, površine...) i koda za NC mašinu na kojoj se dati objekat izrađuje.

Na osnovu ovih primera bi se moglo zaključiti da deklarativno prikazivanje karakteriše prvenstveno CAD, a proceduralno CAM, što ni u kom slučaju nije tačno. Oba načina predstavljanja objekata su podjednako zastupljena u CAD-u, a osobito na polju geometrijskog modeliranja. Na slici 3 je prikazan žičani model trostrane piramide (tetrahedron). Ako ga shvatimo kao skup od četiri temena i šest ivica (slika 3a), u tom slučaju imamo deklarativni način prikazivanja. Međutim, mi isti objekat možemo posmatrati i kao uređeni niz od šest orijentisanih ivica (slika 3b), i u tom slučaju imamo proceduralni način prikazivanja (ovo je lakše shvatiti ako se zamisli buba-mara koja ide po svakoj od ivica i tako opisuje piramidu).

Generalno-posmatrano, deklarativna forma podrazumeva spoljašnje atribute i osobine objekta, kao što su:

- njegove pozicija i orijentacija u odnosu na usvojeni spoljašnji referentni sistem
- globalna geometrija, prikazana u takozvanom „world" koordinatnom sistemu
- unutrašnje prostorne odnose između koordinata tačaka koje pripadaju datom objektu

Prikaz objekta u deklarativnoj formi najčešće mora da zadovolji i geometrijske i topološke aspekte. Ovo se ostvaruje u funkciji takozvanih primitivnih deklarativnih formi i atributa, koji podrazumevaju:

- temena, ivice i površi
- tačke, linije, površine i zapremine
- poziciju i orijentaciju
- jednakost i nejednakost

Primitivne deklarativne forma su u većini slučajeva izražene pomoću algebarske analitičke formulacije.

Slično deklarativnom, proceduralni pristup takođe podrazumeva opis geometrijskih i topoloških aspekata, i uključuje sledeće osobine objekata:

- unutrašnje atribute i osobine, kao što su oblik izražen u odnosu na neki usvojeni unutrašnji referentni okvir
- lokalnu geometriju, datu u odnosu na lokalni koordinatni sistem
- spoljašnje prostorne relacije koje podrazumevaju, na primer, parametarske jednačine ili pravila za relativno pomeranje tačaka, linije ili površine u funkciji određenih parametara

Za formiranje proceduralne forme objekta koriste se sledeće primitivne proceduralne forme i atributi:
- putanje i ciklusi
- pointeri i zapisi
- translacija i rotacija
- algoritmi i procedura

Većina primitivnih formi je izražena korišćenjem iterativne (ili rekurzivne) formulacije.

VERAN PRIKAZ

Već smo nekoliko puta naglasili da, u zavisnosti od atributa koji ga opisuju, svaki prikaz realnog objekta predstavlja u većoj ili manjoj meri samo aproksimaciju. Imajući ovo u vidu, možemo reći da je veran prikaz objekta onaj koji uključuje sve atribute neophodne za uspešan proces dizajniranja i analize. Ovde se podrazumevaju i efekti vrlo realističnog prikaza objekta na displejima radnih stanica. Jedan takav objekat (automobil PEUGEOT 405) urađen pomoću softverskog paketa *Computervision* prikazan je na slici 4.

Postoji više metoda na osnovu kojih se može napraviti geometrijski model nekog tela. U daljem tekstu ćemo navesti tri najčešće korišćena načina za modeliranje, a to su:
- žičani modeli
- površinski modeli
- čvrsti modeli

Na slici 5 je prikazan različit oblik istog poprečnog preseka u zavisnosti od primenjene metode.

U slučaju žičanog modela (Sl. 5a), kutijasti objekat sa slike je prikazan kao skup temenih tačaka i graničnih ivica. Poprečni presek se sastoji od skupa nepovezanih tačaka. Ovakav prikaz je prihvatljiv u slučaju ako nas zanima samo generalni oblik, pozicija ili orijentacija kutije.

Kod površinskog modela (Sl. 5b) kutija je opisana pomoću skupa temenih tačaka, ivičnih linija i njima ograničenim površinama. Poprečni presek je u ovom slučaju definisan temenim tačkama i linijama koje ih povezuju. Ovakvo predstavljanje je pogodno ako nas zanima izgled spoljnih površina objekta, ili ako želimo da ga izradimo iz jednog čvrstog bloka.

Konačno, čvrsti model kutije (Sl. 5c) se sastoji od temenih tačaka, graničnih ivica i površina i njima zahvaćene zapremine. Sa slike se vidi da je poprečni presek sada opisan skupom temenih tačaka, linija koje ove tačke povezuju, kao i zahvaćenim delom površi između ovih linija. Ovo bi bio odgovarajući pristup u slučaju da nas interesuje masa, momenti inercije, dinamičke ili strukturalne osobine. Uz ovo čvrsti modeli se uspešno mogu koristiti za simuliranje interakcije dva ili više objekata, gde su se žičani ili površinski modeli pokazali kao potpuno neodgovarajući. Napomenimo da modeliranje pomoću čvrstih tela zahteva velike memorijske resurse. Ilustracije radi I-DEAS V 5 zahteva minimalno 16 MB RAM-a, a preporučuje se 32 MB, dok diskovi za ozbiljniji rad nebi trebali biti manji od 1 GB.

Međutim, danas je sve manje softverskih paketa u CAD-u koji su zasnovani isključivo na jednoj od prethodno pomenutih metoda. Tako, savremeni CAD modelari omogućavaju direktan prelaz iz jednog u drugi model kako bi se iskoristile prednosti svakog ponaosob. Na osnovu ovog zahteva razvijene su, na primer, „Sewing Operations" – metode kojima se iz postojećeg žičanog i/ili površinskog modela kreiraju čvrsta tela (slika 6).

U sledećem članku iz ove serije više će biti reči o konkretnim matematičkim modelima za predstavljanje objekata i samim modelima.

INTERVJU / CHRIS WELLS

DAN 3D

Na ovogodišnjem, prilično siromašnom, beogradskom Sajmu tehnike, održanom od 13. do 18. maja, nije bilo mnogo pažnje vrednih računarskih firmi. Od onoga što se moglo videti jedan od zanimljivih (možda i najzanimljiviji) je bio štand firme OSA – beogradskog zastupnika Autodeska i par drugih stranih proizvođača grafičkog hardvera. Pored predstavljanja *AutoCAD 11*, *Animatora* i paketa *3D Studio*, na sajmu je boravio i Autodeskov evropski menadžer za multimedia produkte Kris Vels (Chris Wells). Retke trenutke njegovog slobodnog vremena iskoristili smo za kratak intervju.

Računari: Gospodine Wells, Autodesk je trenutno svojim *AutoCAD* i srodnim paketima vladalac na tržištu PC CAD. Imali ste i jedan dosta uspešan pokušaj prodora na tržište 2-D animacije programom *Animator*. Šta vas je nagnalo da, sa *3D Studiom*, krenete u područje vrhunskih programa za fotorealistično prostorno modeliranje i animaciju?

Mr Wells: Paketom *3D Studio* Autodesk ulazi na tržište na kome za sada postoje samo paketi koji su desetak i više puta skuplji. Mi u našem paketu nudimo gotovo sve mogućnosti koje imaju programi reda cena 10.000 i više funti, uz integrisano upravljanje videom kao izlaznim medijem i potpunu dvosmernu DXF kompatibilnost sa *AutoCAD* i ostalim paketima koji prihvataju DXF standard. Pritom je cena oko 20% niža od cena samog *AutoCAD*-a.

Računari: *Animator* je već imao priličan uspeh na tržištu. Jedina ozbiljna zamerka bila je nemogućnost rada u višim rezolucijama od 320 x 200. Zašto niste najpre rešili taj problem?

Mr Wells: Problem nije bio u tome da mi ne možemo napraviti podršku u *Animatoru* za veće rezolucije, već u tome da dosadašnji PC, sve do pojave 486 i brzih VGA, XGA i TIGA karti kao raširenog standarda, jednostavno nisu bili u stanju ni da u realnom vremenu reprodukuju „film" rezolucije 640 x 480 ili više, da ne govorimo o generisanju 3-D animacije u realnom vremenu. Sada je to već moguće, i *Animator* će, pored par drugih poboljšanja, dobiti i mogućnost rada na znatno većim rezolucijama od sadašnjih 320 x 200.

Računari: Kolika bi bila brzina generisanja („rendering") jednog pogleda na naš testni model grada (proširena verzija onog iz prošlog broja – 8000 vektora i 2600 poligona) u VGA ili PAL rezoluciji, odnosno koliko slika za animaciju bismo mogli dobiti za recimo 8 sati neprekidnog rada jednog 486 na 33 MHz sa koprocesorom Weitek 4167 i 8 Mb ili više RAM – ceo program i model su u RAM?

Mr Wells: Pošto vaš testni model ima oko 2600 površina, i verovatno bi imao bar dva izvora svetlosti i par tekstura, mogu vam reći da bi ste na navedenoj konfiguraciji uspeli da „izvučete" preko 2000 fotorealističnih slika tokom osam sati rada, odnosno jedne noći. To vam je 80 sekundi filma, što je, po mom mišljenju, sasvim dovoljno za urbanističku prezentaciju.

Računari: Kako gledate na OS/2 i UNIX kao buduće operativne sisteme na PC i njihovu podobnost za grafičke pakete? Koja PC OS platforma je trenutno najpogodnija za CAD i multimedijalne primene?

Mr Wells: 32-bitni OS/2 i UNIX su daleko pogodniji za ove primene od MS DOS. Međutim, usled relativno slabe raširenosti ovih OS na tržištu, mi za sada sve najpre pravimo za *DOS Extender*. Uz drajvere u 386 zaštićenom načinu on zadovoljava gotovo sve zahteve. Mi sada razmišljamo i o *Windows*-ima.

Računari: A *Cyberspace*?

Mr Wells: Za *Cyberspace* je pokazan ogroman interes, mada morate biti svesni da je za njega potrebna veoma jaka hardverska osnova, najmanje 486 sa dve vrlo brze 3-D grafičke kartice, da ne govorimo o složenosti stereoskopske maske za prostorno prikazivanje. Tek tada ćemo moći da imamo punu simulaciju stvarnosti u realnom vremenu. U bližoj budućnosti ćemo u naš multimedia program takođe uključiti i zvuk, kao i neka druga proširenja.

Nebojša Novaković

IZ MITA U LEGENDU

PostScript je 1985, zajedno sa laserskim printerom, pokrenuo revoluciju u grafičkoj industriji. Dugo godina je predstavljao mit, a danas je legenda.

Robert Slavečki

Da bismo lakše shvatili funkciju PostScripta, i njemu ekvivalentnih programskih jezika najbolje je da se vratimo pet šest godina unazad, u vrijeme kada je PostScript čekao na svoje rođenje. Tih su godina računarskim svijetom dominirala 8-bitna računala popul spectruma, commodore-a, ali i XT kompatibilaca. Pojavljuju se i prvi 16 bitni sistemi, istina više kao slike u časopisima nego kao računala na stolovima. To je ujedno i vrijeme kada kompjutorski giganti očajnički pokušavaju osvojiti poslovne urede i ugurati svoje sisteme poslovnim ljudima.

GRAFIČKI JEZICI

Na Balkanu je stanje nešto drugačije, pojedini entuzijasti sami sklapaju svoje prve ljubimce, kupuju razne „gumice" i tada započinje sumanuto programiranje. Nakon godine ili dvije dočepali ste se i prvog 9 igličnog printera i presretni danonoćno printate svoje programe i divite se svojem djelu. Ali tada postajete zahtjevniji, programiranje vas više ne zadovoljava, pokušavate vaš skupo plaćeni printer iskoristiti za konkretnije stvari, u početku to su najčešće bezlična pisma, a zatim opasno ciljate na poslovne obrasce, referate i slično. Međutim, ova vrste dokumenata obavezno, osim teksta, uključuje i slike, grafikone, i tablice. Tada se suočavate sa nepremostivom barijerom – integracijom teksta i slike, pokušavate nemoguće, tražite pogodne programe, pregledavate mali milion tekst procesora, raspitujete se među kolegama, ali svi su suočeni sa istim problemom. Napokon, zatim odustajete od svega i pokušavate sami doskočiti problemu. Rješenje nalazite u improvizacijama kojih se danas radije ne bi prisjećali.

Istraživači koji su radili na razvoju laserskih printera za uredsku namjenu, bili su davno prije upoznati sa ovim problemom. Shvaćajući da visoko razlučivi laserski printeri nemaju velike šanse na grafičkom tržištu ukoliko se ovo pitanje ne riješi, razvijaju niz algoritama a zatim i jezike kojima je bilo u potpunosti moguće opisati sve grafičke elemente koje je potrebno iz računala poslati na printer. Jednom riječju, razvili su programe koji omogućavaju integraciju teksta i slike zavidne kvalitete. Danas su ti jezici poznati pod imenom PDL (Page Description Language), odnosno jezici za opis stranice. Jedan od najstarijih jezika za opis stranice je i PostScript.

LASERSKA ANATOMIJA

Svaki se laserski printer u osnovi sastoji od dva dijela: ispisnog mehanizma, koji ostavlja otisak na papir, i računala poznatijeg pod imenom RIP (Raster Image Processor).

Ispisni mehanizam laserskih printera radi u tehnologiji koju je razvila kompanija Xerox, a koju danas primjenjuju gotovo svi fotokopirni aparati. Iz RIP-a dolazi regulirani signal (logička 0 ili 1) koji velikom brzinom uključuje ili isključuje lasersku zraku koja se zatim preko zrcala reflektira na rotirajući bubanj. Bubanj je svjetlosno osjetljiv s pozitivnim električnim nabojem. Kada laserska zraka padne na bubanj, odnosna točkica se depolarizira. Bubanj se vrti i na njemu se, točkicu po točkicu, oblikuje grafički sadržaj (stranica). Slika se na bubnju ne vidi dok se na nju ne nanese fini tanki sloj praškastog tonera, koji je također elektrostatski pozitivan. Taj se prašak prima samo na prethodno depolariziranim mjestima, pa slika na bubnju konačno postaje vidljiva.

U sljedećem koraku bubanj dolazi u dodir s papirom, koji je negativno polariziran i zbog toga privlači prah tonera. „Obojeni" papir zatim prolazi kroz rotirajući mehanizam u kojemu se zagrijava na oko 200 C, uslijed čega se tonerski prah zapeče i ostaje trajno učvršćen na papiru. Upravo zbog ovog razloga papir iz laserskih printera, kao i iz fotokopirnih aparata, izlazi topao.

Primarna funkcija RIP-a je da program pisan u nekom od jezika za opis stranice (npr. PostScript), prevede u binarni signal koji će kasnije upravljati laserskom zrakom ispisnog mehanizma. Naime, PDL definira sve elemente stranice kao geometrijske oblike pomoću niza matematičkih formula. Te formule RIP zatim interpretira i iz njih izrađuje odgovarajući raster koji odgovara rezoluciji izlazne jedinice. Dakle, za laserske printere to je najčešće raster u rezoluciji od 300 točaka po inču. Kao i svako drugo računalo, i RIP je izgrađen oko mikroprocesora (najčešće Motorolin 68000), RAM-a u koji se pohranjuje sadržaj stranice za ispis, i ROM-a u kojem je, između ostalog, pohranjen PDL interpreter (npr. PostScript interpreter).

POSTSCRIPT...

PostScript su, 1978 godine u korporaciji Xerox razvili John Warnock i Martin Jewell. Obzirom da nije našao razumijevanje i interes u matičnoj kompaniji, Warnock, zajedno sa još jednim kolegom, 1982. godine osniva vlastitu tvrtku pod imenom Adobe Systems Incorporated. Dalja sudbina PostScript-a svima je manje više poznata.

PostScript je moguće koristiti na dva načina. Većina korisnika upravlja PostScript printerima putem DTP aplikacija kao što su *Corel Draw, Free Hand* ili *Page Maker*. Sa ovakvim je programima moguće na veoma lak način kreirati stranicu i ispisati je na nekom od PostScript printera. Ovi programi sve oblike koje vi nacrtate pomoću miša, definiraju kao vektore i kao takve ih prikazuju na ekranu monitora. Pri slanju ovako definirane stranice na ispis, prethodno odabran PostScript drajver prevodi vektore u PostScript program i takav kod šalje na laserski PostScript printer. Ovaj način korištenja odlikuje se izuzetnom jednostavnošću i ne iziskuje gotovo nikakvo programersko znanje. Iz tog je razloga ovaj način i najrasprostranjeniji i gotovo jedini koji se u stolnom izdavaštvu koristi.

Drugi način predstavlja rad u interaktivnom modu kojim je moguće ostvariti direktnu komunikaciju sa PostScript interpreterom. Ovaj način rada u stvari predstavlja programiranje u PostScriptu koje se može usporediti sa bilo kojim drugim programiranjem, bilo da se radi o bejziku, paskalu ili nekom drugom programskom jeziku.

Najjednostavniji način da se PostScript program pošalje laserskom printeru je pomoću terminala. Kao terminal može, naravno, poslužiti i bilo koje računalo uz odgovarajući komunikacijski program (npr. MacTerminal za Macintosh ili PC-Talk za PC). Rad u interaktivnom modu zahtijeva od korisnika dostatno programersko znanje, ali pruža i bolju kontrolu nad samim PostScript-om. PostScript, naime, sadrži preko 250 naredbi što je izuzetno velik broj. Iz tog razloga niti jedna DTP aplikacija nije u mogućnosti pokriti sve naredbe, a isto tako neke naredbe uopće nije moguće kontrolirati iz grafičkog okruženja pod kojim te aplikacije rade.

Direktno programiranje u PostScript-u u svakom je slučaju prvenstveno namijenjeno programerima koji pišu drajvere za komercijalne DTP aplikacije, ali bi isto tako svaki krajnji korisnik koji te aplikacije koristi trebao znati osnovnu suštinu PostScript-a.

...I PROGRAMIRANJE NA NJEMU

Tri najvažnije karakteristike programskog jezika PostScript su da se interpretira, da se bazira na strukturi staka i da koristi jedinstvenu strukturu podataka zvanu *dictionary*, odnosno u prijevodu 'riječnik'. Riječnik omogućava fleksibilnost i rastezljivost, a činjenica da se PostScript interpretira i da koristi model steka znači da program može biti proizvoljne duljine i složenosti. Obzirom da je za izvršavanje programa dovoljno veoma malo zaglavlje, interpretiranje je moguće izvoditi direktno iz ulaznog sljeda. To ujedno znači da na sam PostScript program nije stavljeno nikakvo memorijsko ograničenje.

PostScript također karakterizira uska povezanost između različitih programskih cjelina. Ta povezanost je veoma utjecajna i korisna jer omogućava da se, modificiranjem koordinatnog sustava, bilo koja stranica jednostavno skalira, rotira ili translatira.

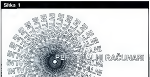
RIJEČNICI I STRUKTURE PODATAKA

Riječnik je struktura koja omogućava organizaciju PostScript programa u potprograme. Svaki se riječnik sastoji od ključa (imena potprograma) te potprogramske cjeline sastavljene od niza PostScript naredbi. Podprogram je moguće imenovati sasvim proizvoljno, i to ime odnosno ključ kasnije, u glavnom programu, koristiti za opoziv dotičnog potprograma. Na ovaj način moguće je sa proizvoljno definiranim imenima kreiranih potprograma manipulirati kao i sa standardnim PostScript naredbama.

Kao primjer možemo razmotriti program *Krug i elipse* u kojem su formulirane dva riječnika, odnosno dva potprograma – *NapraviKružnicu* i *NapraviElipsu*

/NapraviKružnicu ▶

Slika 1




```
% ===== Rotacija teksta =====

/Konturne                    % Stog: string = ---
{ true charpath stroke } def

/Rotacija
  { 15 15 345
    { gsave
      rotate 0 0 moveto
      (Personalni) Konturne
      grestore
    } for
  } def

/HelveticaBold findfont 30 scalefont setfont

250 400 translate
0.5 setlinewidth
Rotacija

0 0 moveto
(PERSONALNI RAČUNARI) true charpath
gsave 1 setgray fill grestore
stroke

showpage
```

Rotacije teksta. Post Script omogućava rotiranje svih grafičkih oblika kao i teksta, kut zakreta se zadaje u stupnjevima.

Slika 4

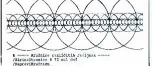



```
% ===== Zumiranje teksta =====
/HelveticaBoldItalic findfont 36 scalefont setfont
/IspišiRačunari
{ 0 0 moveto (računari) show } def
320 400 translate
0.95 -0.05 0 { setgray IspišiRačunari -1 0.5 translate } for
1 setgray IspišiRačunari

showpage
```

Zumiranje teksta. Koristeći osnovne naredbe PostScript interpretera, moguće je kreirati zanimljive slovne oblike.

Slika 2

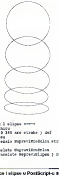


```
% ===== Kružnice različitih radijusa =====
/ŠirinaStranice 8 72 mul def
/NapraviKružnicu
  {XPos YPos Radijus 0 360 arc stroke } def
/PovećajX
  { XPos Radijus add
   /XPos exch def } def
   /LinijeKružnice        % Stog: Radijus Y = ---
     { /YPos exch def
       /XPos 0 def
       /Radijus exch def
       { XPos ŠirinaStranice le
         { NapraviKružnicu PovećajX } { exit } ifelse
       } loop
     } def
10 400 LinijeKružnice
30 400 LinijeKružnice
90 400 LinijeKružnice

showpage
```

Kružnice različitih radijusa. Čitljivost PostScript koda se može povećati ukoliko se logičke cjeline uvuku, ovo je karakteristično za sve više programske jezike

Slika 3



```
% ===== Krug i elipse =====
/NapraviKružnicu
   { 0 0 54 0 360 arc stroke } def
/NapraviElipsu
   { 1 0.75 scale NapraviKružnicu stroke } def

300 500 translate NapraviKružnicu
4 { 0 -72 translate NapraviElipsu } repeat

showpage
```

Krug i elipse. Kružnice i elipse u PostScript-u se definiraju pomoću kružnih lukova.

Ova programska linija, točnije karakter „/", definira početak potprograma i zadaje mu ime, u ovom slučaju *NapraviKružnicu*. Slijedeća linija,

{ **0 0 54 0 360 arc stroke** } **def**

u stvari je proširena naredba za crtanje kružnice. Ovako sastavljen potprogram moguće je kasnije nebrojeno puta pozivati i sa njegovim imenom raspolagati kao i sa svakom drugom naredbom.

PostScript također osigurava i ostale standardne strukture podataka kao što su: stringovi, polja (u koja je moguće smjestiti bilo koji PostScript objekt uključujući i druga polja), cijele brojeve, realne brojeve, imena (koja je moguće davati potpuno proizvoljno navezano uz engleski jezik), i datoteke. Svi ovi tipovi podataka definirani su kao PostScript parametri sa kojima se manipulira u steku parametara. Riječnik može sadržavati bilo koji od navedenih tipova podataka. U PostScriptu funkcija varijable, odnosno promjenjiva striktno je vezana uz riječnik, odnosno za njegovo imenovanje i nema nikakvih drugih funkcija.

MODEL STEKA

Kao što sam već napomenuo, programski jezik PostScript je baziran na stekovima

(stacks). Stek je fiksna struktura podataka u kojoj su elementi pohranjeni privremeno, jedan povrh drugog. Po prirodi, stekovi su takve strukture u kojoj podatak koji je pohranjen zadnji izlazi prvi (Last In, First Out). Ukoliko u stek smjestite tri elementa, zadnji kojeg ste pohranili je na vrhu, i kao takav on je prvi ne dohvatu.

Za razliku od većine drugih programskih jezika u kojima se prvo definira naredba a zatim parametri koji se odnose na tu naredbu, u PostScriptu je odmuto, parametri prethode naredbama. Na primjer, programska linija

216 144 moveto

smješta imaginarni kursor na poziciju 216 horizontalno i 144 vertikalno. Vidimo da su prvo zadani parametri (operandi), a tek nakon toga naredba (operator). Onog trenutka kada interpreter „naleti" na parameter, smješta ga u parametarski stek i tamo ga drži pohranjenog sve dok ne pronađe odnosnu naredbu. Naredba se zatim izvršava, a rezultat se pohranjuje u rezultantni stek. Važno je napomenuti da naredba još uvijek nije interpretirana. Do prevođenja dolazi u onom trenutku kada se u programu pojavi zahtjev za ispis stranice na izlaznu jedinicu. U tom trenutku elementi se izvlače iz stekova, interpretiraju i rezultati šalju na ispisni mehanizam. Ovaj se način prevođenja često naziva i post-fix notacijom.

PostScript interpreter koristi nekoliko stekova – parametarski stek, rezultantni stek, stek rječnika i stek grafičkog stanja. Parametarski stek je jedini u stalnoj upotrebi, dok drugi imaju specijalizirane namjene, te se iz tog razloga rjeđe koriste.

OBLICI I STAZE

U PostScriptu se svi grafički oblici konstruiraju pomoću algoritama koji se nazivaju staze (paths). Staze se sastoje od linija, kružnih lukova ili Bezierovih krivulja. Gotovo je nevjerojatno kako se pomoću ovih osnovnih geometrijskih oblika mogu konstruirati krajnje složeni objekti. Oblik definiran kao staza može se kasnije odjiti aktuelnom bojom ili sivom skalom (rasterom). Staza se također može kopirati, translatirati, rotirati... Ukoliko oblik koji je definiran u stazi zatvara određenu šupljinu, ta se šupljina može ispuniti aktuelnom odjom, rasterom ili nekim uzorkom. Funkciju staze lakše ćete shvatiti ukoliko promotrite kratak program za crtanje sivog kvadrata:

```
newpath
270 360 moveto
0 72 rlineto
72 0 rlineto
0 -72 rlineto
closepath
0.4 setgray
fill
showpage
```

Program započinje naredbom **newpath** kojom se zatvara stara staza (ukoliko postoji), i otvara nova. Nakon toga slijede naredbe za crtanje stranica kvadrata. Naredbom **closepath** zatvara se kvadrat odnosno staza. Slijedeće dvije programske linije služe za ispunjavanje kvadrata sivom bojom. Količina crne odje određuje se naredbom **setgray**, pri čemu je 0 = crna a 1 = bijela. Naredbom **showpage** stranica se fizički ispisuje na pisač.

KOORDINATNI SUSTAV

Već i sama činjenica da je PostScript grafički jezik, čini koordinatni sustav veoma važnom komponentom u programiranju PostScripta. Koliko važnom toliko i problematičnom. Naime, svakom programeru je pojam rezolucije i prikaze grafičkih elemenata na ekran prilično jasan. Kao i ekran, i papir na koji ispisujemo dokument možemo promatrati kao raster, što je ispisna jedinica veće razlučivosti, to je „raster" papira finiji. Problem je upravo u tome što je PostScript nezavisan o razlučivosti izlazne jedinice.

Konkretnije, PostScript program za crtanje kvadrata jednako će raditi na laserskom printeru razlučivosti 300 tpi, kao i na Linotronicu u rezoluciji od 1270 tpi. Jedina je razlika u tome što će na Linotronicu taj kvadrat biti četiri puta manji. Iz tog je razloga u PostScriptu nemoguće koristiti neke fiksne jedinice već se koriste točke čija je veličina relativna i ovisi o razlučivosti izlaza. Ovaj se problem ne ističe toliko u samom konstruiranju objekata koliko u njihovom razmještanju na stranici. PostScript nudi rješenje u vidu dvaju koordinatnih sustava: apsolutnog, koji pokriva cijelu stranicu, i relativnog, koji se odnosi samo na pojedinu stazu kojoj je pridružen. Koliko god se ovo rješenje čini konfuznim, bit će vam jasnije nakon što pogledate sljedeća dva programa:

```
newpath
144 72 moveto
144 360 lineto
stroke
showpage
newpath
144 72 moveto
0 288 rlineto
stroke
showpage
```

Oba programa daju isti rezultat – vertikalnu liniju duljine 4 inča. Razlika je u tome što prvi program koristi samo apsolutni koordinatni sustav u kojem cijeli papir predstavlja jedan, cijeli, apsolutni koordinatni sustav, dok drugi program koristi i apsolutni i relativni sustav. Naredodm **moveto** postavlja se ishodište relativnog koordinatnog sustava na poziciju 144 horizontalno i 72 vertikalno. Parametri dati uz naredbu **rlineto** su relativni i odnose se na relativni koordinatni sustav. Koliko god ovo može djelovati zbunjujuće i beskorisno, relativni koordinatni sustav znatno pomaže kod konstrukcije složenih grafičkih odlika.

POSTSCRIPT FONTOVI

Obzirom da su slova najčešći grafički oblik koji redovno ulazi u svaku grafičku publikaciju, svi PDL jezici polažu veliku pažnju upravo na fontove. Adobe je tokom razvoja PostScripta uspio razviti nekoliko arhitektura kojima se definiraju slovni oblici, a danas je najpopularniji format pod imenom Type 1. Važno je napomenuti da se PostScript fontovi odnosno slova po svojoj konstrukciji ni po čemu ne razlikuju od ostalih grafičkih oblika kao što su kružnice, linije i krivulje. Jedina je razlika što se fontovi gotovo redovito koriste češće od bilo kojeg drugog odlika. Iz tog razloga fontovi su spremljeni u posebne pakete – datoteke, koje je prema potrebi moguće pozivati i koristiti. Organizacija fontova unutar paketa i sama konstrukcija fonta unutar PostScripta upravo je poznata pod nazivom Type 1. Ovakva arhitektura fontova omogućava nam da uz standardne grafičke komande, koristimo i specijalne naredbe koje se odnose isključivo na tekst.

```
/Times findfont
15 scalefont
setfont
72 200 moveto
(računari) show
show page
```

Priloženi program ispisuje tekst „računari" na poziciji 72 horizontalno i 200 vertikalno. Tip fonta kojim je tekst ispisan je Times, a veličina je 15 točaka.

Iako Adobe, i ostale digitalne slovolivnice u svijetu, ali i kod nas, nude cijeli spektar različitih fontova, moguće je kreirati i vlastite slovne oblike koristeći posebne font editore kao što su *Fontographer* ili *FontStudio*. Fontovi se također mogu kreirati i direktno u PostScriptu iz postojećih grafičkih oblika. Ovaj se postupak međutim ne preporuča – jednak je skakanju kroz prozor sa drugog kata na glavu.

POSTSCRIPT-UM

Stolno izdavanje je danas visoko komercijalizirani posao kreiranja i printanja dokumenata, od knjiga i časopisa pa čak i do dnevnih novina. Odgovarajući računalni sistemi se danas i kod nas sve intenzivnije koriste za ubrzavanje i olakšavanje nekad mukotrpnog slovoslaganja, grafičke pripreme i tiska.

PostScript je ponudio sustavno rješenje za probleme dizajniranja i tiskanja dokumenata. Njegova najveća prednost je neovisnost o rezoluciji i izlaznoj jedinici. Iz tog razloga moguće ga je koristiti na laserskim printerima za uredske potrebe, kao i za visoko zahtjevnu produkciju na fotoosvjetljivačkim jedinicama poput Linotronica L300. Obzirom da sve grafičke oblike definira pomoću matematičkih formula, moguće je svaki element stranice (ili cijelu stranicu), povećavati, smanjivati, translatirati, rotirati i vršiti cijeli niz drugih transformacija. Uz napomenuto, PostScript sadržava i niz naredbi kojima je moguće ubacivati i sve vrste bit mapira na grafike uključujući i rasterske slike, bez kojih je nezamisliva svaka grafička produkcija. Uza sve ove egzotične poslove, PostScript se može koristiti i za jednostavnije operacije kao što je npr. hard kopija ekrana.

Na žalost, ili na sreću, postoji i druga strana medalje. Najveća mana PostScripta je njegova nepristupačnost koja je prvenstveno rezultat snoodvske politike koju je Adobe vodio sve do nedavno. Naime, svaki proizvođač koji je koristio PostScript i ugrađivao ga u svoje jedinice morao je postotak od prodaje davati Adobe-u. Zamislite koliko bi zaradio npr. Microsoft da je ovu politiku primjenjivao samo na svoj C kompajler. Svaki programer koji bi koristio C morao bi Microsoftu davati proviziju. Međutim Adobe se mogao ovako ponašati dok je pomoću PostScripta držao monopol, i dok je PostScript predstavljao neku vrstu unikatnog rantela. Ali danas...

Godine 1984. PostScript je pokrenuo revoluciju. Bio je prvi, bio je najbolji, bio je revolucionaran. Međutim, svima je poznato da svaka revolucija pokreće i evoluciju. Usporedo sa razvojem PostScripta pojavljivali su se i drugi jezici za opis stranica i stvarali nove standarde. Istina, PostScript je i dalje držao primat, ali staklenih nogu. Te su se noge u potpunost razbile kada je Apple prije nešto više od godinu dana objavio novu tipografsku arhitekturu za računale pod nazivom *Truetype*. Apple-u se u razvoju pridružio i Microsoft i bitka je započela. Rezultat toga je da Adobe, da bi se koliko toliko održao u poslu, određene programske pakete kao *Adobe Type Manager* dijeli gotovo besplatno.

Pogrešno bi bilo pretpostaviti da će ova situacija izbaciti PostScript iz računarskog svijeta. Naprotiv, konkurencija će ga učiniti pristupačnijim i jeftinijim nego što je to ikada bio. Jedini gubitnik je trenutno – Adobe.

NOVEL UZVRAĆA UDARAC

Tek što smo pisali o *LAN Manager*-u, o tome kako Microsoft namjerava ostvariti ideju korporativnog umrežavanja, o tome kako mu to za sada ne polazi za rukom, i o tome kako to ipak namjerava učiniti u skoroj budućnosti (uz malo pomoći od prijatelja), Novel povlači svoj slijedeći potez i ponovo ovladava situacijom u centru.

Dragutin Vuković

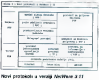
Najnovijom verzijom *NetWare* operativnog sistema Novell je uspostavio jaku vezu s UNIX platformom osobnih računala i tako načinio veliki korak naprijed, prema ostvarenju ideje korporativnog umrežavanja. *NetWare/386 V3.11*, pušten u prodaju krajem prvog kvartala 1991. godine, uključuje i na strani servera i na strani klijenta podršku za slog protokola TCP/IP (od engl. *Transmission Control Protocol / Internet Protocol*), standardni slog za UNIX sisteme. Serveru je dodana i podrška NFS (od engl. *Network File System*), skupu distribuiranih programa za podršku sistema datoteka razvijenom u Sun Microsystems Inc.

INTEGRISANA ARHITEKTURA

Ovim korakom Novell je opet iskoračio jednom nogom ispred svog starog rivala Microsofta, u vječitoj utrci za bolji mrežni operativni sistem. Microsoft tek treba dodati podršku TCP/IP-a, NFS-a ili OSI-ja svom *LAN Manager*-u. *NetWare 3.11* također je premostio i pukotinu koja je razdvajala Novell od Banyan Systems, koji svojim operativnim sistemom *Vines* odavno podržava vezu sa UNIX svijetom.

Dajući podršku TCP/IP i NFS, *NetWare 3.11* omogućuje korisniku daleko veće mogućnosti komunikacije kroz mrežu i preko međumrežnih veza. Novi pristup umrežavanju Novell je nazvao *Integrisanom računarskom arhitekturom* (ICA, od engl. *Integrated Computing Architecture*). ICA definira veze, podatke, pouzdanost, osiguranje i upravljanje mrežom. Implementirana je kao modularni sistem programa takozvanih NLM (od engl. *NetWare Loadable Module*), koji se mogu povezati s jezgrom operativnog sistema. Držeći pojedine usluge i aplikacije u odvojenim modulima, Novell omogućuje korisniku da instalira samo one usluge koje najviše odgovaraju njegovim potrebama. Modularnost također omogućuje da jezgra operativnog sistema ostane stabilna prilikom dodavanja novih modula i izmjena postojećih.

TCP/IP slog protokola u *NetWare 3.11* radi istom brzinom i na istoj razini performansi kao i *NetWare*-ov osnovni slog protokola IPX/SPX (od engl. *Internet Packet Exchange / Sequenced Packet Exchange*). Napisan je u asembleru tako da može upotrebljavati standardni skup



Novi protokoli u verziji *NetWare 3.11*

sistemskih rutina *NetWare*-a. Novell je isprobao svoju implementaciju TCP/IP sloga protokola na Sun Microsystems Conectathon konferenciji prošle godine, komunicirajući s desetcima raznih sistema. Korisnici koji su ga vidjeli na djelu bili su impresionirani.

Greg Scott, upravnik odjela za računarske usluge na Oregonskom državnom univerzitetu u Corvallisu kaže da je PC server pod *NetWare 3.11* operativnim sistemom nadmašio performansama Sun-ov *Sparcserver* vrijedan 200 000 - USD, koji se također nalazio u mreži. Mreža na kojoj je iskušan novi *NetWare* server sadržavala je PC-e, Sun i Next radne stanice pod UNIX-om, IBM RS/6000 radne stanice i Macintoshe. Scott također kaže da su na univerzitetu bili iznenađeni i veoma zadovoljni performansama Novell servera, te da je pristup datotekama bio veoma brz i transparentan za korisnike.

USMERAVANJE PAKETA

Novellov TCP/IP slog protokola dinamički usmjerava IP pakete koristeći protokol RIP (od engl. *Routing Information Protocol*), a također može upotrebljavati i statičko usmjeravanje gdje korisnici unaprijed određuju putanje paketa u tablice za usmjeravanje. S tim mogućnostima usmjeravanja paketa korisnici mogu komunicirati preko međusobno povezanih mreža sa udaljenim UNIX sistemima, drugim računalima na TCP/IP mreži ili udaljenim mrežama pod *NetWare 3.11*. U ovom posljednjem slučaju TCP/IP protokol obavlja funkciju poznatu kao „tuneliranje" na odašiljačkoj strani se IPX paketi ekapsuliraju unutar TCP/IP paketa i šalju u mrežu. Na prijemnoj strani se uklanja enkapsulacija TCP/IP paketa i IPX paket isporučuje u svom prirodnom obliku *NetWare* serveru.

TCP/IP paketi se mogu usmjeravati kroz više mrežnih segmenata koji mogu biti sličnih ili različitih topologija: Ethernet, Arcnet ili Token Ring, zahvaljujući Novellovom otvorenom podatkovnom sučelju ODI (od engl. *Open Datalink Interface*). ODI odvaja gornje slojeve TCP/IP sloga protokola od fizičke implementacije mreže, adaptera i upravljačkih programa, te implementira mehanizam razmjene paketa među različitim slogovima protokola i različitim upravljačkim programima za mrežne adaptere (Slika 1.).

Novell 3.11 sadrži i komplet razvojnih alata za kreiranje aplikacija. Komplet uključuje biblioteke za dinamičko povezivanje (DLL, od engl. *Dynamic Link Library*) za sučelje transportne razine (engl. *transport level interface*) Berkeley Sockets, i pozivanje udaljenih procedura (RPC, od engl. *Remote Procedure Call*) s odgovarajućim sučeljima za aplikacione programe. Korištenjem ovih sučelja u aplikacijama, moguće je kreirati aplikacijske programe koji će raditi i sa *NetWare* i sa UNIX računalima, koristeći TCP/IP.

NetWare 3.11 omogućuje korisnicima *NetWare*-ovog sistema za razmjenu poruka (MHS, od engl. *Message Handling System*) da razmjenjuju poštu sa korisnicima UNIX-baziranog jednostavnog protokola za razmjenu pošte. Također je implementiran i protokol za jednostavno upravljanje mrežom (SNMP, od engl. *Simple Network Management Protocol*).

IZMEĐU DOS-a i UNIX-a

Modul mrežnog sustava datoteka (NFS, od engl. *Network File System*) implementira sustav datoteka koji je „domaći" za korisnike UNIX sistema, a također im omogućuje i korištenje mrežnih printera. Korisnici PC-a u mreži mogu, zahvaljujući NFS-u, razmjenjivati datoteke s korisnicima UNIX-sistema i dohvatiti datoteke na UNIX računalima u mreži. *NetWare* NFS ima za sada jedan nedostatak – nema svoj dio na strani klijenta. Stoga korisnici ne mogu direktno sa svoje DOS stanice pristupiti UNIX sistemu datoteka, već to moraju učiniti posredstvom *NetWare* servera. U Novellu kažu da će se i taj dio razviti ako potražnja od strane korisnika to opravda. (NFS podrška za klijenta može se dobiti kod nekih nezavisnih proizvođača, npr. SK-PC-NFS od Schneider & Koch-a). NFS nije uključen u cijenu operativnog sistema, kupuje se odvojeno i košta oko 5 000 - USD.

Podršku TCP/IP sloga protokola na strani klijenta za pojedinačne stanice pruža Novell



LAN Workplace za *DOS/Windows* što znači da *NetWare* klijenti ne moraju koristiti usluge servera za ostvarivanje TCP/IP veze sa UNIX sistemima već to mogu učiniti direktno – čak i kad je *NetWare* server van pogona. Kombinacija TCP/IP sloga protokola i ODI upravljačkog programa mrežne kartice ne zauzima više od 45 KB u memoriji klijentove stanice. Uz upotrebu programa za upravljanje memorijom, na 386 baziranim sistemima, većina TCP/IP protokola može se prebaciti u proširenu memoriju i tako još smanjiti zauzeće konvencionalne memorije.

LAN Workplace uključuje tri TCP/IP aplikacije. Predstavljač računala domaćine (engl. *Host Presenter*) omogućuje korisniku da ima čak deset istovremenih Telnet veze sa raznim računalima. FTP server (engl. *Serving FTP*) je server protokola za prijenos datoteka koji radi u pozadini ispod MS *Windows*. „Ekspres datoteka" (engl. *File Express*) je *Windows* program koji pojednostavnjuje prijenos datoteka između klijentove stanice i UNIX sistema. Alati za *LAN Workplace* sadrže DLL-e za *Berkeley Sockets* i *NetBIOS*. *LAN Workplace* košta 450,- USD za jednokorisničku verziju, a 1.995,- USD za deset korisnika.

JOŠ NEKA POBOLJŠANJA

Uz poboljšano povezivanje sa sistemima drugih platformi i operativnih sistema, *NetWare 3.11* nudi još neka poboljšanja nad prethodnom verzijom. Jedna od njih je i sučelje ka IBM-ovom *NetView* programu za upravljanje i nadzor mreže. Ovo sučelje omogućuje *NetWare*-ovim ugrađenim uslugama upravljanja mrežom da šalju poruke i izvještaje *NetView* upravljačkoj konzoli. *NetWare* također može slati podatke *NetView*-u putem IBM SNA veze uspostavljane preko IBM računala domaćina. *NetWare 3.11* sadrži i novi uslužni program SBACKUP. Ovaj program omogućuje administratorima mreže da prave zaštitne kopije (backup) i restauriraju podatke na više servera odjednom, kao i da restauriraju NetWare univerzalni sistem datoteka.

Neki rani korisnici *NetWare 3.11* žele se da novim programima za nadzor i upravljanje mrežom nedostaju neka svojstva koja je imao stari program FCONSOLE iz *NetWare 286*. Novi RCONSOLE ne daje informacije o upotrebi „cache" spremnika za diskove, niti statistiku performansi sistema. Novel obećava uskoro poboljšanja ovih usluga.
da

Novost u verziji 3.11 je i to da je Novell na ovu verziju primijenio svoju poznatu politiku podrške korisnicima sa skromnijim zahtjevima koju je razvio na NetWare 286. Dok je u NetWare/386 verzijama 3.0 i 3.1 postojala samo varijanta za 250 korisnika koja je bila prilično skupa, verzija 3.11 dolazi u tri gradacije: za 20 korisnika (3.495,- USD), za 100 korisnika (6.995,- USD) i za 250 korisnika (12.495,- USD).

I na kraju – šećer. Svi korisnici *Novell NetWare* verzija 3.0 i 3.1 dobijaju besplatno proširenje na verziju 3.11 ako to zatraže do 30.09.91. To se, naravno, odnosi samo na legalne korisnike koji su svoje operativne sisteme nabavili preko ovlaštenih prodavaca i uredno su registrirani.

Sve što vas zanima u vezi verzije 3.11 i ostalih verzija Novell NetWare možete saznati na adresi: MicroLAB d.o.o. Karela Zahradnika 26, 41020 Zagreb, tel. (041)692-703, fax. (041)692-707

BRZINOM SVETLOSTI

Tvrdi diskovi i, uopšte, sva druge disk memorija za smeštaj velikih količina podataka predstavljaju najuže grlo svakog računarskog sistema. Rešenje ovog problema nije u stalnim tehničkim poboljšanjima, nego u krajnje radikalnim tehničkim zahvatima.

Danas je već svima postalo jasno da je najveće ograničenje, i najveći problem u daljem ubrzavanju rada računala kao cjeline, upravo brzina rada disk jedinica. Proizvođači gotovo svakodnevno nude razna poboljšanja u brzini ovih komponenata u vidu novih protokola, novih standarda, boljih kontrolera itd. Međutim, sve su ovo kratkoročna rješenja – rješenja kojima se ne rješava sam bit problema. Naime, današnje disk memorije (npr. tvrdi diskovi) su mehanički uređaji i zbog toga su ograničene u brzini okretne ploče. Te ruke za pisanje i čitanje pokretane elektromotorom nikada se neće moći približiti brzini kojom elektroni „jure" kroz procesor i ostale „čisle" elektroničke komponente.

Na sreću, prema najnovijim podacima, ovaj problem neće biti vječan. Istraživači rade na razvoju 3-D memorija i holografskoj tehnici pohranjivanja podataka kojima bi se izbjegli mehanički problemi prisutni kod postojećih disk jedinica. Nova rješenja temelje se na optici, odnosno pohranjivanju podataka svjetlom.

Iako laserske zrake već igraju značajnu ulogu u CD-ROMovima, ovi optički diskovi još uvijek zahtijevaju mehaničke dijelove pokretane elektromotorima kojima se prelazi preko površine diska i na taj način očitava svjetlosni zapis. Nova tehnologija neće imati niti jednog jedinog pokretnog mehaničkog dijela.

Na Kalifornijskom sveučilištu u Irvinu, profesor kemije Peter M. Rentzepis uspješno eksperimentira sa (kako ih on naziva) „trodimenzionalnim memorijama". Rentzepis koristi lasersku zraku za zapisivanje podataka na fotoosjetljiv preparat ugrađen u plastičnu kocku. Ultraljubičasti laser usmjeren pod odgovarajućim kutovima, na različitim stranama kocke, simultano ispaljuje lasersku zraku, i na mjestima gdje se zrake sijeku, fotoosjetljiv materijal absorbira svjetlost. Nakon što dvije UV zrake pogode molekulu fotoosjetljivog preparata, dolazi do slabljenja međumolekulske veze. Svaka skupina ovakvih molekula predstavlja jedan bit informacije.

Za čitanje podataka, spomenuta se kocka gađa zelenim laserskim zrakama. Zeleno svjetlo pobuđuje molekule čije su veze oslabljene, izazivajući emitiranje crvene svjetlosti. Fotodioda upija zatim tu svjetlost i generira električni impuls koji se zatim šalje na BUS računala. Za brisanje podataka, sustav opet koristi drugu valnu frekvenciju. Rentzepis razmišlja već sada o propuštanju laserske zrake kroz leće od tekućeg kristala. Kombinacijom adekvatnih električnih impulsa pridruženih tekućem kristalu, mogao bi se kontrolirati fokus leća. Na taj način eliminirala bi se potreba za pokretnim dijelovima.

Bez pokretnih dijelova, 3-D memorije bile bi izuzetno brze. Vrijeme pristupa današnjih tvrdih diskova mjeri se u milisekundama (tisućinkama sekunde), gdje je vrijeme od 18 ms izuzetno brzo. Rentzepis predviđa da bi vrijeme pristupa 3-D memorijskog sustava kretalo u granicama od 9 do 10 nanosekundi. Zapisivanje u 3-D memorije bilo bi još brže. Uz ovakva predviđanja, brzina računala ne bi više bila ograničena memorijama već procesorom.

Kao što je impresivno vrijeme pristupa 3-D memorija, jednako je nevjerojatan i podatak o kapacitetu tih memorija. Sustav bi teoretski dozvoljavao pohranjivanje 1250 gigabajta u plastični komad nešto veći od kockice šećera. Rentzepis sumnja da će 3-D memorije ikada doseći teoretska predviđanja, ali prema njemu bi bilo moguće praktički izraditi kocku kapaciteta od oko 130 gigabajta!

U istraživačkom konzorciju MCC (Microelectronics and Computer Technology Corporation) u Austinu (Teksas), istraživači su došli do rješenja za pohranjivanje podataka koje nije u tolikoj mjeri dramatično kao i 3-D memorija, ali je znatno bliže praktičnoj realizaciji. Osnovana od strane nekolicine tehnoloških korporacija, MCC je razvila mogućnost korišćenja lasera za pohranjivanje podataka u holograme.

Kao i kod 3-D memorije, i ovdje se koristi intenzivno svjetlo kako bi se izazvale promjene u strukturi kristala. Za razliku od Rentzepisove kocke, ovaj se medij zasniva na holografskim kristalima položenim na ravnu podlogu. Za kreiranje dodatne gustoće zapisa moguće je ovako dobivene ploče nanositi jednu na drugu, te na taj način na istu površinu postaviti nekoliko holograma. MCC se nada da će uspjeti proizvesti medij za pohranu podataka veličine i oblika diskete od 5.25" i kapacitetom od 100 MB. U sljedećoj generaciji kapacitet bi se povećao do 1 GB.

Prednost MCC-ove memorije nad već postojećim disketnim medijima očituje se u brzini. Steve Redfield, direktor razvojnog programa u MCC-u, predviđa vrijeme pristupa od 10 mikrosekundi i vrijeme zapisa od 100 mikrosekundi, što je oko tisuću puta brže od današnjeg najbržeg tvrdog diska. Jedan od razloga zbog kojeg holografske memorije rade na ovako velikoj brzini je i činjenica da one zapisuju i čitaju podatke simultano, dakle cijelu skupinu podataka odjednom.

Priroda holografa također daje novom mediju izvanredan stupanj obimnosti i sigurnosti podataka – predmet pitanja koji je od velike važnost za svakoga koji je ikada izgubio vitalne informacije zbog tehničke greške memorijske jedinice. Dovoljno je napomenuti da holografske memorije nisu osjetljive na statički elektricitet, razna električka pražnjenja, utjecaj magnetskog polja, radijaciju, mehaničke utjecaje.

Iako su 3-D memorije okarakterizirane boljim svojstvima i ciljaju na znatno duži vremenski period, holografske memorije su znatno bliže stvarnosti, konkretnije tržištu. MCC radi već pune tri godine na razvoju prototipa, a konzorcij upravo započinje sa drugom, dvogodišnjom, fazom u kojoj bi trebali proizvesti radni model memorije. MCC neće sam proizvoditi novi proizvod, već će proizvodnju povjeriti članovima kompanija koje su potaknule i osnovale istraživanja. Na njima ovisi kada će se holografske memorije pojaviti na tržištu, a po prvim se procjenama nagađa da to neće biti u sljedećih nekoliko godina. Čak da se i ove memorije pojave za recimo dvije do tri godine, prvi bi primjerci bili instalirani u specijalnim računarskim sistemima. Nakon toga, trebalo bi još nekoliko godina da se uspostave novi standardi i tek bi se tada ove memorije mogle pojaviti na širokom tržištu.

Robert Slavečki

NIKO KAO ETERNET

U ovom broju prikazujemo osobine i usporedne testove jedanaest Ethernet adaptera najpoznatijih svjetskih proizvođača. Broj je nešto manji od obećanog, jar svi testovi nisu stigli na vrijeme. Testiranje je obavljeno u National Software Testing Laboratories (NSTL), Plymouth, USA. NSTL je nezavisna organizacija koja se bavi testiranjem osobnih računala i lokalnih mreža.

Dragutin Vuković

ISPITNA MREŽA

Evo kako je bila izgrađena ispitna mreža

- Compaq Deskpro 386/20 server pod Novell NetWare 286 V2.15 operativnim sistemom
- Compaq Deskpro 386/25 primarna stanica (na njoj su izvođeni glavni testovi i mjerene performanse)
- 16 x Compaq Deskpro 286 sekundarne stanice (za opterećanje mreže prometom i testove sa višestrukim adapterima)
- Compaq Deskpro 386/20 kontrolna stanica (za automatiziranje testova)

Kao medij je upotrebljen tanki Ethernet koaksijalni kabel (RG 58A/U). U kontrolnu stanicu i 13 sekundarnih stanica instalirani su adapteri 3Com EtherLink Plus za vrijeme čitavog testiranja. Ispitivani adapteri instalirani su u server, primarnu stanicu i tri sekundarne stanice

Aplikacija koja je generirala pozadinsko opterećenje mreže načinjena je pomoću *Foxbase/LAN V2.1*, izvodila je oko 30 transakcija u minuti. Svaka transakcija pregledavala je indeksirane zapise u tri datoteke, zaključavala, mijenjala i otključavala po jedan zapis u svakoj datoteci.

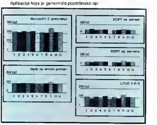
Mrežni operativni sistem je regeneriran za svaki adapter, koristeći upravljačke programe za adaptere koje isporučuje proizvođač adaptera. Mrežni programi za stanice također su generirani s upravljačkim programima proizvođača adaptera. Stanice s adapterima koji nisu testirani (13 sekundarnih i kontrolna stanica) imale su iste adaptere i upravljačke programe za vrijeme cijelog testiranja.

TEST APLIKACIJA

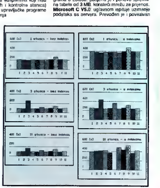
Štampanje u serverski red (slika 1.) ispituje sposobnost adaptera da obavlje uobičajene mrežne poslove. Tekst datoteka veličine 660 KB kopirana je na preusmjereni priključak štampača LPT1. Izlaz je naredbom CAPTURE preusmjeren u serverski red za štampanje. Mjereno je vrijeme između pokretanja naredbe za štampanje i povratka u DOS.

Lotus 1-2-3 simulira svakodnevnu upotrebu elektroničkih tabela u mreži. Ispituju se neke od više mrežno orijentiranih aktivnosti koje korisnici redovito koriste. Upotrebom makro programa u Lotusu punjena je i ponovo spremana tabela od 3 MB, koristeći mrežu za prijenos.

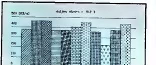
Microsoft C V5.0 uglavnom ispituje uzimanje podataka sa servera. Prevođen je i povezivan



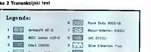
Slike 1 Test aplikacija



Slika 2 Transakcijski test



Slika 3 Test propusnosti IPX protokola



veliki program u XLISP 2.0 C izvornom kodu (cca 12000 redaka, 250KB) i kreiran izvršni program od cca 150 KB.

XCOPY na server testira prenošenje podataka na server. Kopirano je stablo direktorija (13 grana, 130 datoteka) na server naredbom XCOPY /S. Izvorno stablo nalazilo se na disku primarne stanice.

XCOPY sa servera ispituje prenošenje podataka sa servera. Prenošeno je isto stablo direktorija sa servera na disk primarne stanice.

OBRADA TRANSAKCIJA

Ovaj skup testova (slika 2.) intenzivira tranzijentne aktivnosti u mreži, tj. generira veliki broj kratkih paketa sa zahtjevima serveru. U testu je upotrebljeno pet datoteka pod *Foxbase/LAN V2.1*. Svaka transakcija čita iz prve datoteke, pronalazi i zaključava zapise u sljedeće tri datoteke, dodaje zapis u petu datoteku sa zaključavanjem, mijenja sve zaključane zapise i zatim otključava sve zapise.

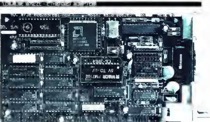


Tabela 1.: Ocjene i cijene mrežnih adaptera	perfor-manse	svoj-stva	upotreb-ljivost	ukupna ocjena	cijena [000]
Racal Interlan NI6510 Data Link	8.7	6.6	8.8	8.5	495
Western Digital Ethercard Plus16	8.5	6.0	6.6	8.3	699
Artisoft LANtastic AE-2	6.8	4.3	6.5	6.4	369
DCC Ixolan 4110-2	8.4	6.4	8.6	8.0	605
CNet CN010B	7.3	4.0	8.8	8.5	439
Gateway G/Ethernet	8.6	5.8	6.4	8.1	445
Novell NE2000	8.8	6.6	6.8	7.4	495
Pure Data PDI8023-16	—	6.0	9.1	—	495
SMC Ethernet-PC510	8.8	7.6	7.6	6.8	886
3Com EtherLink Plus	5.7	8.6	8.6	8.6	895
Tiara LanCard/E*16	8.7	6.6	6.6	8.6	895

Tabela 1. Ocjene iz pojedinačnih kategorija i ukupna ocjena za sve testirane adaptere, dodana je i cijena adaptera, radi usporedbe

Test obrade transakcija izveden je u šest faza. U prvoj fazi je jedna stanica (sekundama) obradila 960 transakcija. U drugoj fazi su tri stanice obradile svaka po 320 transakcija bez preklapanja tj. nisu pristupale istim zapisima. U trećoj fazi je šest stanica obradilo po 960 transakcija svaka. U prve tri faze nisu upotrebljavani indeksi, dok su u sljedeće tri ponovljani isti testovi, ali sa upotrebom indeksa.

Indeksiranje je usporilo obradu jer povećava aktivnost diska i takmičenje za pristup datotekama.

Tabela 6.: Pregled svojstava testiranih mrežnih adaptera				
mrežni adapter	sučelje s računalom	upravljački i ostali programi proizvođača adaptera	podrška mrežnih operativnih sistema	dodaci
NI6510 Data Link	16-bita gospodar sabirnice	dijagnostika, NetWare 686, NetWare 386, NDIS, NCSA Telnet	IBM LAN Server, NetWare 686, NetWare 686, 3Com 3+Open, TCP/IP	Boot PROM za NetWare [$50]
Ethercard Plus16	16-bita dijeljena memorija	dijagnostika, NetWare 286, NetWare 386, 3Com 3+Share, DOS, DECNet, DOS/PCSA	Banyan Vines, IBM PC LAN, NetWare 686/686, 3+Open, 3+Share, TCP/IP	LattisNet primopredajnik [$156]
LANtastic AE-2	16-bita programirani ulaz / izlaz	NetWare 686, NetWare 386, LANtastic	NetWare 686, NetWare 386, Artisoft LANtastic	Boot ROM [$88], 64 KB dod. memorija [$58]
Ixolan 4110-6	16-bita gospodar sabirnice	dijagnostika, NetWare 286, NetWare 686, PC-NFS/NFS, PC/TCP, Net3100, OS/2 SB	Banyan Vines, IBM PC LAN, NetWare 686/186, 3+Open, IBM LAN Server, TCP/IP	NEMA
CN010B	16-bita programirani ulaz / izlaz	nema dijagnostike	NetWare 686, NetWare 286,	terminatori ugrađeni na adapteru
G/Ethernet	16-bita programirani ulaz / izlaz	dijagnostika	NetWare 286, NetWare 386, 3+Share, TCP/IP	Boot ROM
NE2000	16-bita programirani ulaz / izlaz	nema dijagnostike, NetWare 286, NetWare 686	NetWare 686, NetWare 386	NEMA
PDI8023-16	16-bita dijeljena memorija	dijagnostika, NetWare 686	NetWare 686, NetWare 386	model za 10BaseT
Ethernet-PC510	16-bita programirani ulaz / izlaz	dijagnostika, NetWare 686, NetWare 386, program za konfiguriranje	Banyan Vines, IBM PC LAN, NetWare 686/386, 3+Open, IBM LAN Server, TCP/IP	UNIX, NetLine, PC/TCP, WIN/TCP, Boot ROM
EtherLink Plus	16-bita programirani ulaz / izlaz	dijagnostika, 3+Open, LinkPlus optimizator	Banyan Vines, NetWare 286, NetWare 386, 3Com 3+Open, 3Com 3+Share, TCP/IP	128 KB dodatne memorije
LanCard/E*16	16-bita programirani ulaz / izlaz	dijagnostika, SCO 889IX, Interactive UNIX, Net3100, DECNet-DOS	LAN Server, NetWare 686, NetWare 386, 3Com 3+Open, 3Com 3+Share, TCP/IP	Boot PROM za NetWare [$50]

Tabela 2. Sažeti pregled svojstava mrežnih adaptera, uključujući sučelje prema računalu, programsku podršku koja se dobija uz adapter (dijagnostika, upravljački programi, pomoćni programi itd.), te popis mrežnih operativnih sistema i protokola s kojima adapter može raditi.

TEST PROPUSNOSTI IPX-a

Ovaj test (slika 3.) mjeri propusnost kartice i njenog upravljačkog programa u KB/s prenoseći datoteku veličine 3MB između dvije stanice pomoću IPX transportnog protokola. Podaci su prenošeni između dva Compaq 386/25 računala u prvoj fazi. U drugoj fazi je aktiviran i prijenos iste datoteke između dva Compaq Deskpro 386s računala, tako da su se oba prijenosa odvijala istovremeno. Podaci su prenošeni u blokovima veličine 512 bajtova. Upravljačko programi adaptera nezavisno su određivali veličinu stvarnog Ethernet paketa za prijenos. Rezultati iz obadvije faze za sva četiri sistema kombinirani su da se dobije prosječna vrijednost propusnosti.

U ovom testu pokazale su se najveće razlike među performansama adaptera i njihovih upravljačkih programa. To, međutim, nije sasvim realistična mjera, iako ovaj test dobro mjeri ponašanje adaptera. Razlog tome je statična veličina bloka od 512 bajtova i nesudjelovanje servera u prijenosu, što nije uobičajena situacija u mreži.

OCJENE

Stručnjaci iz NSTL, nakon provedenog testiranja, ocijenili su sve testirane adaptere prema tri kategorije: performanse, svojstva i upotrebljivost. Ocjena performansi proizašla je direktno iz rezultata testova, tj. predstavlja ocjenu sposobnosti adaptera da obavlja uobičajene svakodnevne zadatke. Svojstva adaptera ocijenjena su na osnovi kriterija podrške mrežnim operativnim sistemima, različitim prijenosnim medijima, višestrukim protokolima, standardima i dijagnostičkim programima. Na ocjenu upotrebljivosti adaptera najviše je utjecala njegova fleksibilnost pri instaliranju i konfiguriranju. Nedovoljan izbor prekidnih linija, U/I i memorijskih adresnih područja veliki je nedostatak, ali ni veliki izbor nije prednost ako je konfiguriranje komplicirano. Uz sve to, naravno, neophodna je i razumljiva i iscrpna dokumentacija. Ukupna ocjena dobijena je težinskom sumom ocjena iz sve tri kategorije, s težinama: performanse 7, svojstva 1, upotrebljivost, 2.

DEZINFICIRAJTE MREŽU!

Postojanje računarskih virusa činjenica je koju više ne možemo i ne smijemo zanemarivati. Pojave „epidemija" računarskih virusa mnogo su češće nego što se o tome čuje; u većini poduzeća ne govori se javno o pojavama virusa iz straha da se ne okrnji ugled firme i ugrozi poslovanje. Računarski virusi šire se najviše u okolinama u kojima se mnogo razmjenjuju ilegalne kopije programa, oblik ponašanja koji lako možemo nazvati „računarski promiskuitet".

Dragutin Vukavić

ZLOČESTI PROGRAMI

Destruktivni programi raznih vrsta (vidi okvir „Mala enciklopedija zločestih programa") imaju jedno zajedničko svojstvo: jednom kad se aktiviraju, izvršavaju funkcije s kojima korisnik nije upoznat i koje nije želio izvršiti. Ove funkcije mogu biti samo dosađujuće i ometajuće, a mogu uzrokovati i gubitak, modificiranje ili uništavanje podataka i programa i degradaciju performansi sistema. Glavni problem je u tome da ih korisnik ne može kontrolirati: oni se razmnožavaju, uvlače se u druge programe i aktiviraju se kad im se to svida. Čak i kad sami ne rade štetu, svojom aktivnošću ometaju rad i umanjuju performanse računala. O vrstama „zločestih programa", načinima njihovog razmnožavanja i borbi protiv njih pisano je već u Računarima i drugdje, pa ovdje nećemo ulaziti ponovo u te detalje. Također, radi jednostavnosti sve vrste ovih programa u ovom ćemo tekstu nazivati zajedničkim imenom virus.

Virusi, dakle, postoje i veselo se razmnožavaju po našim diskovima. Ne morate se zbog toga uplašiti. Zaražavanje računala je još uvijek rijedak događaj. Također, relativno jednostavnim akcijama može se samostalno računalo efikasno zaštititi od infekcije ili izliječiti.

Sasvim druga priča je kada se računalo nađe u mreži. Ako virus zarazi računalo koje je uključeno u mrežu i povezano bar s jednim serverom, gotovo je sigurno da će virus ubrzo preći na server. Sa servera će vrlo lako zaraziti sve stanice koje se prijave na taj server. Pošto se stanice često prijavljuju na više različitih servera, zaraza će se proširiti i na druge servere, proći će preko mostova u druge mreže i „epidemija" je tu. Ako „dezinficiramo" zaraženu stanicu, ona će se ponovo inficirati čim se prijavi na server. Dezinficiramo li i server, on će se zaraziti čim se neka zaražena stanica na njega prijavi.

Kao i kod bioloških epidemija, za suzbijanje epidemije računarskih virusa nije dovoljno samo raspolagati vakcinom. Potrebna je dobro organizirana i koordinirana akcija da se najprije zaustavi širenje zaraze, zatim unište uzročnici i najzad saniraju štete. Upravo o tome i jeste riječ u ovom tekstu: kako spriječiti prodor virusa u mrežu i kako ih iz nje protjerati ako ipak uspiju ući. Naglasak je pri tome baš na organizaciji preventivnih i sanitativnih akcija, a ne na samim sredstvima i alatima za otkrivanje i uništavanje virusa. Kako je, ipak, potrebno znati čime raspolažemo u borbi za zdravlje naše mreže, dat ćemo pregled vrsta programskih proizvoda koji nam u tome pomažu.

ANTIVIRUSNA ALKEMIJA

U posljednjih nekoliko godina na tržištu (i van njega) pojavio se priličan broj programskih paketa koji služe za zaštitu, otkrivanje i uklanjanje virusa iz računala. Njihovi proizvođači, naravno, tvrde za svakog od njih da potpuno štiti računalo od svih napasnika. Istina je da to niti jedan ne može učiniti. U momentu dok kupujete program koji će vas sigurno zaštititi od svih trenutno u svijetu postojećih virusa, netko, negdje, već priprema novu vrstu virusa od koje vas taj program neće zaštititi. Kao i u biološkoj medicini, niti ovdje ne postoji „panacea" (lijek za sve bolesti). Osim toga, što veći stupanj zaštite pruža neki antivirusni proizvod, tim više vas ometa i sputava u radu. Program koji svaki pristup disku predusretne i traži od korisnika dozvolu za obavljanje funkcije bit će strahovito nepraktičan i korisnik će se ubrzo odreći takove zaštite. (Ako već ne možete normalno raditi na svom računalu, najbolje ga ugasite. Tako nećete raditi ništa, ali je računalo stopostotno zaštićeno od virusa.)

Za borbu protiv invazije virusa koriste se tri tipa programa, koji često dolaze zajedno u paketu. Programi za preventivu infekcije pokušavaju spriječiti virus da prodre u računalo. Načinjeni su obično kao rezidentni programi koji provjeravaju svaki izvedivi program prije nego se on počne izvoditi u memoriji računala. Jedna metoda je da se svakom izvedivom programu na disku odredi kontrolna suma i zapiše u posebnu datoteku. Kasnije kad se program treba pokrenuti, ponovo se izračuna kontrolna suma i usporedi s prethodno zapisanom. Ako je u međuvremenu došlo do infekcije, kontrolne sume će se razlikovati. Ova metoda neznatno usporava pokretanje programa, ali je komplicirana za održavanje jer se svaki novi program mora registrirati sa svojom kontrolnom sumom u popisu ispravnih suma. Druga metoda u svakom izvedivom programu, prije nego se on pokrene, traži takozvani „potpis" virusa. Potpis je jedinstveni niz znakova karakterističan za određeni tip virusa. Zaštitni program sadrži potpise većeg broja poznatih virusa. Pronađe li i jedan od potpisa u izvedivom programu, neće dozvoliti izvođenje tog programa i tako će spriječiti aktiviranje virusa. Ova metoda je jednostavnije ali manje pouzdana od prethodne. Virus čiji potpis nije registriran nesmetano će ući u računalo. Stoga je kod ovih programa potrebno neprestano ažuriranje liste virusa.

Programi za detekciju virusa, nazvani virus skanerima, najčešće se koriste za potvrdu postojanja virusa kad se već pojavi sumnja. Oni također rade na principu traženja potpisa, ali nisu rezidentni iako da ni malo ne utječu na performanse sistema. Ako se pojavi sumnja da je u sistemu virus, ili tek onako radi provjere, pokrene se virus skaner koji pretraži sve datoteke na disku u potrazi za nekim od potpisa sa piste. Inficirane datoteke se prijavljuju. Prilično su efikasni u otkrivanju poznatih virusa, ali im također treba stalno dopunjavati listu potpisa. Tehnika uspoređivanja kontrolnih suma i ovdje može vrlo dobro poslužiti. U posljednje vrijeme su se pojavili neki novi virusi nazvani virusima polejnicima (engl. *stealth virus*). Ovi virusi mogu detektirati rad virus skenera i prevariti ga. Sam virus uzima s diska datoteku, uklanja iz nje zarazni kod i predaje ju skaneru na provjeru. Skaner, naravno, ne otkriva virus i vraća datoteku na disk pri čemu ju virus ponovo zarazi. Zbog toga neki skaneri pretražuju i memoriju računala u potrazi za takvim virusima. Redovita, periodička primjena virus skanera može doprinijeti ranom otkrivanju virusa prije nego on počne raditi svoje nepodopštine.

Programi za dezinfekciju u stvari bi se trebali nazivati programima za kirurgiju datoteka. Oni u stvari pokušavaju iz zaražene programske datoteke izrezati kod virusa i datoteku dovesti u stanje prije infekcije. Pri tome se može desiti da odrežu previše, što će dovesti do neispravnog rada programa, ili premalo, što može dovesti do daljnjeg širenja virusa. Najpouzdanija tehnika za rekonstrukciju izvedivih programskih datoteka je čuvanje originalnih, zdravih programa na disketama zaštićenim od upisivanja. Zaražene datoteke na disku treba uništiti i iskopirati sa disketa originalne programe. Pri uništavanju zaraženih programa treba obratiti pažnju da nije dovoljno samo pobrisati datoteke. Naredba DEL samo uklanja deskriptor datoteke iz tablice direktorija. Sadržaj datoteke, uključujući i infektivni dio koda, ostaje na disku, iako se do njega normalnim putem kroz operativni sistem ne može doći. Međutim uslijed neke sitne greške u nekom programu može doći do ponovnog aktiviranja virusa. Stoga je prije brisanja potrebno preko sadržaja datoteke upisati neke bezopasne znakove, npr. nule.

ŠTO UČINITI DA SE NE DOGODI

Poznata poslovica „Bolje spriječiti nego liječiti" vrijedi i u ovom slučaju. Najbolji način borba protiv zaraze mreže virusima je da se spriječi ulaz virusima u mrežu. Slijedi sedam savjeta kako to postići.

1. Obrazujte korisnike.

Objasnite korisnicima što su virusi, koji su tipični simptomi njihove pojave. Odštampajte i podijelite brošure s uputama kako se zaštititi od virusa. U ovim uputama svakako treba uključiti slijedeće:

- uzdržavati se od upotrebe ilegalnih kopija programa;
- programe koji su u slobodnoj razmjeni (engl.: *shareware*) nabavljati od autora ili ovlaštenog BBS-a;
- nabavu i instaliranje novih programa prepustiti ovlaštenim i stručnim osobama.

Korisnik koji svakodnevno radi na svom računalu najbolje će ocijeniti kada se dogodi neka sumnjiva promjena u računalu. Korisnike treba opskrbiti popisom tipičnih znakova pojave virusa: promjena veličine, datuma i/ili vremena programskih datoteka, degradacija performansi sistema, pojava nepoznatih datoteka, neočekivan nedostatak memorije za izvođenje programa, neočekivan gubitak prostora na dis-

ku nestajanje datoteka, neuobičajene poruke i znakovi na monitoru, pojava većih segmenata loših sektora na disku.

Vjerojatno je nemoguće forsirati striktno pridržavanje svih tih pravila i preporuka barem ne bez formiranja neke vrste programske policije. Najbolja metoda je da se uvjeri korisnike da pridržavanjem ovih pravila štite sebe od gubljenja rezultata svog (i tuđeg) rada.

2. Pravite zaštitne kopije

Najvažnije u borbi protiv virusa je minimizirati štete nastale njihovom pojavom. Viruse u principu nije teško uništiti, ali je teško oporaviti se od štete koju oni nanose. Najbolji način za smanjivanje štete je da se redovno i to što češće prave rezervne kopije podataka sa servera i ostalih računala na kojima se nalaze važni podaci. Važno je pri tome voditi računa da se na rezervne kopije upisuju samo podaci, jer se kopiranjem programa može na zaštitnu kopiju prenijeti i virus. Originalne diskete s programima i njihove radne kopije treba čuvati posebno, zaštićene od upisivanja. U slučaju pojave virusa programe treba restaurirati s originalnih disketa, a ne sa zaštitne kopije. Ako se sumnja na pojavu virusa, može se usporediti veličine programskih datoteka na disku s originalima.

3. Kontrolirajte administratorove prava pristupa mreži i serverima

Administratori mreže imaju opasnu naviku da svom korisničkom imenu, kojim se služe za svakodnevni rad, dodijele sva prava pristupa. Tu je potencijalna rupa u sigurnosti, jer ako se njegovo računalo zarazi u opasnosti je čitava mreža. Administrator bi trebao imati dva korisnička imena, jedno sa ograničenim pravima za svakodnevni rad i jedno sa proširenim pravima za održavanje mreže.

4. Kontrolirajte korisnička prava pristupa mreži i serverima

Ako je moguće ne dozvolite korisnicima da mijenjaju atribute programskih datoteka na serveru do kojih imaju pristup. Ograničite korisniku pristup samo na neophodna područja na disku. Što mu je manje programskih datoteka dostupno, manje ih može zaraziti.

5. Skenirajte servere

Diskove na serverima treba redovito skanirati da se otkrije prisutnost virusa. Što je češća upotreba virus skenera prije će se otkriti virus i biti će manje štete. Proces se može automatizirati kod mreže sa mogućnošću vremenskog raspoređivanja izvođenja programa. Stanica sa koje se izvodi program za skaniranje serverskih diskova mora biti najprije provjerena da sama nije zaražena.

6. Zaštitite stanice

Mrežne stanice treba zaštititi od infekcije. Za to može poslužiti neki rezidentni detektor virusa, ali to nije najpouzdanije rješenje. Bolje rješenje je kontrola ulaza novih programa. Virusi se najviše šire na disketama s igrama. Ako se korisniku onemogući unošenje igara (i općenito novih programa) bez prethodne provjere od strane administratora, uveliko se smanjuje mogućnost zaraze (i povećava produktivnost). Dobre mrežne stanice imaju ugrađene funkcije koje podržavaju ovaj način borbe protiv viruse (vidjeti članak o mrežnim stanicama, *Računari* br. 71): kontrolu pristupa disketnom pogonu, kontrolu punjenja operativnog sistema i sl. Uz upotrebu dobrog sistema izbora (menu-a) može se izvesti da se na određenoj mrežnoj stanici mogu pokretati i izvoditi samo oni programi koje omogući administrator mreže.

7. Nadgledajte instaliranje novih programa

Neki komercijalni programski paketi, kupljeni kao novi, u originalnom pakovanju, bili su zaraženi virusima. Najvjerojatnije je to bila posljedica politike nekih prodavaca programa da nude takozvanu garanciju „vraćanje para" (engl. *money-back guarantee*), kojom se omogućuje kupcu programa da u određenom roku vrati kupljeni program, ako nije njime zadovoljan, i dobije natrag svoje novce. Prodavac raspolaže alatom kojim takav programski paket ponovo zapakira u plastični omotač i prodaje ga kao novi. Ako je prethodni kupac inficirao diskete nekim virusom, slijedeći kupac bit će neugodno iznenađen. Iako se to danas izuzetno rijetko događa, jer su i prodavci opskrbljeni alatima za detekciju virusa, nije na odmet novi program najprije instalirati „u karantenu", tj. na posebnom računalu, odvojenom samo za tu namjenu, ili barem odvojenom od mreže.

ŠTO UČINITI KAD SE DOGODI

Pojava virusa u mreži jednaka je pojavi epidemije neke opasne bolesti. Stoga pravila i upute za borbu protiv virusa nužno sliče pravilima za suzbijanje epidemije. Prije nego što se virus pojavi treba pripremiti plan za akciju u slučaju pojave virusa. Za kreiranje ovog plana, a i njegovo provođenje u djelo, treba odabrati „krizni štab", grupu sastavljenu od stručnjaka iz raznih odjela poduzeća koji su svi vezani za upravljanje, održavanje, razvoj i upotrebu računarske mreže. Plan akcije treba svakako sadržavati slijedećih sedam koraka.

1. Okupiti krizni štab

Nakon što je detektirana pojava virusa treba što prije okupiti krizni štab i započeti provođenje plana za borbu protiv virusa. Važno je napomenuti da virus treba biti pouzdano detektiran, da se ne diže panika zato što je netko nepažnjom obrisao dio vlastitih datoteka pa pomislio da je to djelo virusa. Upotreba virus skanera pomoći će u tome, međutim ako skaner ne detektira virus to može značiti i da se radi o novoj vrsti virusa koju skaner ne prepoznaje. Jedan drugi način detekcije je da se virusu dozvoli da se sam otkrije: stavi se nekoliko kontroliranih „zamoraca" u okolinu na koju se sumnja da je zaražena. Ako se i oni zaraze – dokaz je tu. Postupak je slijedeći: stanica se pokrene sa „sterilne" diskete, iskopira se nekoliko programa „zamoraca" sa originalnih, pouzdano nezaraženih disketa, instalira se antivirusni paket koji koristi registracijsku metodu i odrede se kontrolne sume programa „zamoraca", pokrenu se programi koji su pod sumnjom da su zaraženi, provjere se kontrolne sume „zamoraca" da se utvrdi jesu li zaraženi.

2. Proglasite obustavu rada

Prva akcija u suzbijanju epidemije je da se zaustavi njeno daljnje širenje. U slučaju računarskih virusa to znači prestati aktivirati zaražene programe, dakle zaustaviti svaki rad u napadnutoj mreži. Obustava rada mora trajati sve dok krizni štab ne zaključi da je infekcija uspješno suzbijena.

3. Procijenite nastalu štetu

Krizni štab mora ustanoviti koliko se virus raširio, odnosno kolika je šteta učinjena. Započnite od jedne zaražene mrežne stanice, najbolje od one na kojoj je najprije zamijećena pojava virusa. Ustanovite primarni server kao i sve druge servere na koje se korisnik te stanice prijavljuje. Ti serveri se moraju smatrati zaraženim sve dok se ne dokaže suprotno. Upotrijebite virus skaner ili metodu kontrolnih suma na svim mrežnim diskovima tih servera. Ako se na serveru pronađe virus, tada se zaraženima moraju smatrati i sve stanice u segmentima mreže povezanim s tim serverom. Određivanje opsega „epidemije" kritična je operacija za uspjeh čitavog plana. Ako se greškom podcijeni ovaj opseg, virus će se ponovo pojaviti. Ako se zaraza precijeni, rezultat će biti nepotrebni gubici radnog vremena.

4. Obavijestite sve korisnike koji su došli pod utjecaj virusa

Sve korisnike za koje je u prethodnoj fazi ustanovljeno da su, možda, došli u dodir s virusom, treba obavijestiti o zarazi te o akcijama koje poduzima krizni štab. Njihove mrežne stanice treba fizički odspojiti od mreže da ne bi prenijeli virus na nezaražene servere ili ponovo zarazili tek dezinficirani server. Stanice treba isključiti jer, iako su odvojene od mreže, virus se može širiti na lokalnom disku i načiniti daljnju štetu na njemu. Ako korisnici mogu elektronskom poštom prenositi programske datoteke, treba obavijestiti sve korisnike pošte da ne koriste tim putem dobijene programe.

5. Izolirajte zaražene mreže i servere

Čim se na serveru otkrije prisustvo virusa, svi korisnici koji su povezani s tim serverom moraju se odvojiti od njega, ugasiti stanice i fizički ih odspojiti od mreže. Odmah treba na serveru zabraniti daljnji pristup korisnicima. Samo jednoj stanici treba dozvoliti pristup, onoj sa koje će se pokrenuti dezinficiranje i restauriranje servera. Ta stanica i sama mora prethodno biti dezinficirana.

Ostali serveri u istoj mreži ne moraju se isključivati, ali ih sve treba provjeriti. Ako je mreža povezana s drugim mrežama, npr. mostovima, potrebno je isključiti sve te međumrežne veze. Nakon dezinfekcije, server mora ostati izoliran sve dok se sve zaražene stanice ne dezinficiraju.

Programske datoteke na serveru treba rekonstruirati sa originalnih disketa.

6. Sanirajte mrežne stanice

Niti jedna stanica se ne smije ponovo spojiti u mrežu dok se pouzdano ne utvrdi da na njoj više nema virusa. Inficirane programske datoteke treba restaurirati sa originalnih disketa. Ni ovdje se ne preporuča upotreba programa koji uklanjaju viruse iz programskih datoteka.

7. Spojite stanice u mrežu

Nakon što su virusi uklonjeni sa svih zaraženih servera i stanica, stanice se mogu ponovo spojiti u mrežu, na serverima se može dozvoliti pristup korisnicima a mogu se uključiti i međumrežne veze. Ako je postupak suzbijanja „epidemije" dobro proveden mreža će opet normalno raditi a virus se više neće pojaviti. Ako je bilo gdje u mreži ostala makar samo jedna kopija virusa, uskoro će se sve ovo ponoviti.

MALA ENCIKLOPEDIJA ZLOČESTIH PROGRAMA

Trojanski konj: Naizgled koristan program koji obavlja i poneku neočekivanu funkciju, izvan onih koje je autor opisao u upotrebnoj dokumentaciji. Funkcija se može pokrenuti kad se steknu neki uvjeti (određeni datum i vrijeme, n-to pokretanje programa, kada popunjenje diska postigne određeni postotak kapaciteta i sl.). Skup uvjeta koji pokreće funkciju naziva se „okidačem", a sama neočekivana funkcija naziva se „bombom" ili poklonom.

Crv: Program načinjen tako da koristi i manipulira slabosti i greške u operativnom sistemu. Ima sposobnost da se razmnožava i na taj način uspori, zaguši i najzad zaustavi računalo. Crv ne sadrži „bombe".

Virus: Virus je zarazni trojanski konj. Ima sposobnost da se razmnožava i da se „prikači" na druge programe, zarazi ih. Virus iz zaraženog programa može se dalje širiti. Kao i trojanski konj virus skoro uvijek ima i skrivenu „bombu", izuzetno opasan u mreži.

KAD BI SE VEZALI NA JUPAK

Kada je krajem 1989. kreiran Sezam, bio sam gotovo zapanjen brzinom kojom su Zoran Životić i Dejan Ristanović ostvarili svoju „prijetnju" da će napraviti BBS na bazi vlastitog softvera. Pri ozbiljnim BBSovima je vlastiti softver svakako najbolji izbor, no uz uvjet da ima tko da ga napiše. Njih su dvojica bili baš kao stvoreni za to. Do tada su već dobro poznavali BBS-ove, vrline i mane drugih programa, a da o vještini programiranja posebno i ne govorimo. :-)

Darko Bulat & Andrea Maja Fajdiga

I tako je Sezam otvoren na jednoj telefonskoj liniji. Program je bio napisan za IBM PC pod DOS-om. Koncepcija se bazirala na stavu „jedan korisnik, jedan procesor". Razmatrala se i druga moguća koncepcija, tj. BBS pod operativnim sistemom UNIX. No, UNIX još u to doba nije bio previše raširen, pa tako niti dovoljno poznat, da bi se pod njim mogao napisati novi BBS program.

Odabrana koncepcija je vodila u onom smjeru u kojemu je danas Sezam duboko zagazio, prema kojem proširenje BBS-a ovisi o broju dostupnih telefonskih linija, te o dodatnom hardveru, jer se svaki korisnik priključivao doista na vlastiti fizički PC, koji je bio povezan s ostalima putem Novell mreže. I tu su počeli problemi tipa „Dokle tako?" i „Kako dalje?".

ŽELJE...

Želje su da Sezam može opsluživati i dalje čim više korisnika. Pri tome, određeni bi broj korisnika pristupao Sezamu kao i do sada direktnim biranjem telefonskog broja Sezama, iza kojeg se nalazi mala sanja od šest linija. Vrlo je vjerojatno da će ih u budućnosti biti i više. Za usporedbu, najveći „mali" BBS na svijetu, popularni EXEC-PC ima sada već 230 dolaznih telefonskih linija. U budućnosti bi direktno zvali uglavnom oni korisnici koji gravitiraju Beogradu. Ostali bi koristili JUPAK.

Stoga je jedna od želja i vezivanje Sezama na našu paketsku mrežu JUPAK. S druge strane, lokalna mreža po kojoj teku podaci između čvorova Sezama ima ograničen kapacitet. Kako broj čvorova u mreži raste, raste i ukupan promet u njoj, mreža postaje sve opterećenija, da bi se jednog trena dostigao gornji limit proširenja mreže, poslije čega mreža više ne služi ničemu. Isto tako je želja da se resursi ugrađeni u Sezam, tj. diskovi, CPU snaga, memorije i slično, koriste što optimalnije.

Odgovor na ova pitanja ovisi o pretpostavljenoj koncepciji.

... I MOGUĆNOSTI

Ako BBS proširimo i dalje, prema sadašnjoj koncepciji, snaga jednog čvora još će uvijek biti ekvivalentna snazi jednog XT-a ili AT-a, što ovisi o izboru čvornog računala. Čvorni XT ili AT zauzima prostor, prosječno troši oko 2.5 KWh struje na dan, čak ponešto i „prča" kroz mrežu, bez obzira da li je trenutno korisnik na njemu ili nije. Ako jeste, onda je stanje redovno, no ako nije, tada je šteta i te struje, neiskorištene CPU snage i memorije što se tu nalazi. Međutim, već sada se nude na prodaju „kompjuteri na pločici". Na jednoj kartici pune dužine nalazi se jedan ili više XT-a ili AT-a. Više takvih kartica stavlja se u jedno kućište, a od „matičnog kompjutera" se koristi ispravljač, te komunikacija preko kompjuterske sabirnice. Takva međusobna komunikacija automatski smanjuje opterećenje lokalne mreže. Na taj način su već neki resursi optimalnije iskorišteni.

S druge strane, UNIX mašine bazirane na 486 procesoru imaju snagu procesora koja višestruko nadmašuje onu od običnog XT-a ili AT-a, a pod UNIX-om je ta snaga optimalno raspoređena na korisnike. Dok neki korisnik vrši *download*, drugi editira poruku, treći čita priloge u konferencijama, a nekolicina međusobno razgovara u *chat*-u, raspodjela I/O kapaciteta, diska ili CPU a nikako nije jednolika. Istovremeno, za akciju ekstrahiranje podataka iz konferencije i njihovu kompresiju, korisnik na AT-u će uvijek imati brzinu (sporiju) AT-a, dok će na 486/UNIX-u moći koristiti punu snagu 486 procesora, ako se tom tipu procesa dodijeli visok prioritet. Dakle, prvi razlog za UNIX je optimalna raspodjela resursa. Ne moram posebno isticati da je danas moguće sagraditi 486-icu sa 16 Mb glavne memorije, sa 700 Mb diskom, sa superbrzim caching disk kontrolerom poput HyperStore ili DPT koji na sebi može imati megabajte memorije za posredovanje zapisa između glavne memorije i diska, s „inteligentnom" RS-232 karticom poput DigiBoard, Specialix ili Anvil, koja podupire brzine do 38400 bps po kanalu u istom trenu. Takva mašina je izuzetno snažna na I/O operacijama, a to je upravo ono što je za jedan BBS potrebno.



Pocelo je na na jednom AT-u 12 MHz pod DOS-om (na slici autor osnovna aplikacije Zoran Životić)

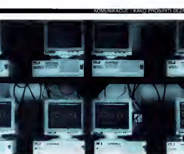
Kad jednog trena snaga 486-ice više nije dovoljna, možemo birati između umrežavanja više takvih računala pomoću TCP/IP mrežnog protokola i nabavke snažnije UNIX mašine. Prva je varijanta vjerojatno jeftinija, jer se dokupljivanjem hardvera onaj stari ne baca. No, s programske strane, ta je varijanta nešto složenija, jer bi neke dijelove kode valjalo proširiti kako bi međuprocesna komunikacija uz oznaku korisnika i datoteke imala i oznaku mašine (npr. sezam1, sezam2, sezam3...). Tako bi, recimo, korisnik „pero" na mašini „sezam1" bio kasnije „pero@sezam1", a datoteka „/dir1/file2" na mašini „sezam2" bi bila „sezam2:/dir1/file2". U biti, trebalo bi samo uključiti standardne TCP/IP mehanizme za protok informacija kroz mrežu. Na žalost, ne možemo napraviti usporedbu s kombinacijom DOS/*Novell*, jer *Novell* mreže ne poznajem dovoljno.

Druga je mogućnost za proširenje nabavka snažnijeg računala pod UNIX-om. Za ilustraciju, konferencijski sistem magazine Byte, popularni BIX, je danas baziran na superračunalu „Pyramid" koji je hlađen tekućim helijem. BIX ima jako mnogo telefonskih linija, rasprostranjenih širom SAD, no nisu sve te linije BIX-ove. Pa kako je to napravljeno?

JAVNA MREŽA

Evo ovako. U SAD postoji nekoliko javnih X.25 mreža. X.25 mreža je takva kompjuterska mreža, gdje svako priključeno računalo ima jedinstvenu brojčanu adresu. Kako su podaci koji teku takvom mrežom razdijeljeni u manje pakete, obično do 128 bajta, takva se mreža naziva još i paketskom. Komunikacija između računala odvija se slično komunikaciji među običnim telefonskim pretplatnicima. Neko računalo, nazovimo ga računalo A, pomoću posebnog programa, odabere broj računala B, te na taj način paketskoj mreži izrazi zahtjev da želi komunicirati s računalom B. Ukoliko je računalo B takvog tipa da može primiti poziv, te ukoliko je doista i u stanju da primi poziv, tada računalo B, taj poziv i primi. Od onog momenta kada je veza uspostavljena, podaci mogu teći, a vlasnik mreže može zaračunavati minute „razgovora" i količinu podataka koji proteku. Po isteku komunikacije, bilo koje od računala može poslati zahtjev za raskid veze. Tu je jedna od razlika između tog tipa komuniciranja i običnog telefoniranja, gdje samo strana koja naziva, tj. strana A, može zahtijevati raskidanje veze.

Treba napomenuti da su računala u X.25 mrežu koja se u Jugoslaviji zove JUPAK, povezana stalnom fizičkom vezom, tzv. iznajmljenim vodovima, te je ta usluga u startu mnogo skuplja od telefoniranja. Prednost je što podaci



a nastavilo se na standardnih 6 (u jednom trenutku i na čitavih 15) linije pod Novell-om na principu jedna veza - jedan računar, da li je sledeći korak u razvoju sisteme prelazak na UNIX?

teku bez greške, što su moguće brzine i preko 9600 bps npr 19200 i 48000 te što je međugradska komunikacija kroz JUPAK dosta jeftinija od telefoniranja Eto, zbog toga bi Sezam želio na X 25. No da nisam još objasnio što bi običan smrtnik imao od toga da se Sezam veže na JUPAK. Tim više što je za mnoge taj JUPAK posve nedostupan Polako strpljen – spašen

Pretpostavimo, dakle, da je Sezam na neki način vezan na JUPAK te da se pozivanjem njegove adrese dobije poznati Sezamov identifikacijski prompt Već to će biti teško postići, ali i o tome kasnije Od tog trenutka pa nadalje, komunikacija teće posve identično onoj kad direktno nazovemo Sezam Izborom komande „bye", Sezam će, kao računalo B inicirati raskid veze Za one koji imaju svoj račun na nekom od VAX/VMS računala vezanih na JUPAK, uspostavljanje veze ne bi trebalo predstavljati problem To bi se obavilo pomoću komande „set host /x29 <SEZAMova_adresa>"

Međutim, za one kojima se ne naziva direktno Beograd telefonom jer je skupo, a ipak bi željeli biti povremeno na Sezamu postojala bi mogućnost da Sezam za svoje pretplatnike kupi JUPAK NUI To je posebna identifikacijska oznaka, koja omogućava da se koristi JUPAK putem telefona a da se opet zna tko je inicirao poziv te da se tako i tarifira. No, bilo bi preskupo da se kupuje NUI za svakog korisnika Sezama posebno Dovoljno bi bilo da ih ima tek nekoliko te da se oni korisnicima dodjeljuju nasumično, ali i to tek prema zahtjevu NUI bi bio takvoga tipa da ne bi dozvoljavao nazivanje bilo čega drugoga, osim Sezama te bi se na taj način onemogućile zloupotrebe Taj bi se tip NUI-a koristio tako da se nazove JUPAK-ov telefonski broj u Zagrebu ili Ljubljani sačeka se prompt „>", te se na njega ukuca nešto poput ovoga „> NBLA01/KVA23,A " Pri tome su BLA01 i KVA23 dva dijela NUI-a koji se ne vide odk se kucaju, te dakle predstavlja ju tajnu Ono A na kraju označava da želimo uspostaviti vezu prema onom računalu čija je adresa unaprijed uprogramirana uz dati NUI Drugim riječima pomoću datog NUI-a bi bilo moguće zvati isključivo Sezam, te prema tome tarifiranje ne bi bilo teško obaviti i iz Sezama

OBRAČUN TROŠKOVA

Kada bi korisnik pristupio Sezamu Sezam bi prvo trebao utvrditi s koje je fizičke linije došao poziv Ako je u pitanju jedna od telefonskih linija, tada se poziv ne tarifira obzirom da korisnik sam snosi troškove direktnog poziva u Beograd Ako je poziv došao kroz JUPAK treba utvrditi s koje adrese A je nastupio poziv Ako je nastupio s nekog drugog računala u JUPAK-u ili čak neke inozemne paketske mreže tada opet ne treba vršiti tarifiranje jer korisnič ke strana sama plaća i taj poziv Ako se međutim utvrdi da poziv dolazi korištenjem nekog NUI-a čiji je vlasnik Sezam, a tada valja pokrenuti mehanizam za obračunavanje troškova kako Sezam ne bi bio na gubitku Sezam za NUI i za korištenje JUPAK-a plaća PTT-u, korisnik Sezama plaća Sezamu i stvar je riješena Naravno, sve ovo ima smisla jedino ako je ukupan zbroj novaca što ga mora korisnik Sezama izbrojati Sezamu za korištenje NUI-a i telefonski račun za lokalni poziv ako u mjestu postoji ulaz na JUPAK ,npr Zagreb ili Ljubljana, manji od iznosa što bi ga korisnik platio kada bi samo zvao direktno Beograd Ne želji da se prođe jeftinije, a da usluga ipak bude dobra se i bazira čitava ova priča Eh da Putem NUI-a bi trebalo dozvoliti pozive jedino registriranim korisnicima ali ne i onima koji dolaze samo testirati Sezam jer takve pozive ne bi bilo moguće tarifirati Naime za pretpostaviti je da će NUI-i vremenom postati javna tajna

Na veoma sličan način ovoma je omogućeno korisnicima BIX a širom USA da pristupaju BIX-u a da ne moraju nazivati njegov direktan telefonski broj BIX je priključen na X 25 mrežu TYMNET, a njegova je puna međunarodna X 25 adresa +310690157800 U Jugoslaviji se nemjesto znaka „+" stavlja „6" dok je u drugim zemljama to „0"

> Andrea Maja Fajdiga je postdiplomand fizike čvrstog stanja na Institutu Jožef Stefan u Ljubljani Na Sezamu se može naći kao "endrea" a na DECNET-u kao CATHY FAJDIGA Darko Bulat je sistemski administrator na Institutu Ruđer Bošković u Zagrebu Na Sezamu se pojavljuje kao "dbulat", a na DECNET-u kao OLIMP BULAT

KAKO NA JUPAK?

Dobro, savladali bismo vještinu pozivanja Sezama iz drugih mjesta no to bi ionako predstavljao lakši dio problema Sada je na redu posljednji i najsloženiji dio bez kojega nema Sezama na JUPAK-u Odgovor na ovo pitanje će ovisiti u najvećoj mjeri o krajnjem izboru tehnologije samog Sezama Ako se, naime ipak krene u pravcu „SEZAM na UNIX-u" tada je rješenje gotovo trivijalno jednostavno uz uvjet, naravno da se dopiše dio softvera za tarifiranje No, kako je tarifiranje potrebno u oba slučaja onda je ovo rješenje „čišće" od drugih mogućnosti Postoji nekoliko opcija, počev od korištenja ALTOS 486-ice koja standardno nudi vezu na X.25 mrežu pa od korištenja gotovih hardversko-softverskih rješenja za priključenje bilo koje druge UNIX/486 mašine na X 25 mrežu Takva rješenja dodaju UNIX-u nove virtualne linije Uz postojeće fizičke linije „tty01 tty02 ..." koje bi bile vezane na modeme za direktne pozive, došle bi nove linije „ttx01, ttx02 ", koje bi odgovarale pojedinim virtualnim kanalima X 25 veze U samom softveru ne bi trebalo mijenjati ništa jer je UNIX-u svaka „ttXXX" linija jednako vrijedna

Ako se međutim nastavlje s koncepcijom „jedan korisnik jedan CPU" tada se stvari kompliciraju Iako ne treba smetnuti s uma da bi prijelaz s jedne koncepcije na drugu samo po sebi bio dovoljno složen posao ipak bi valjalo pažljivo odmjeriti koja je koncepcija dobra na duge staze Dakle ako uzmemo da svaki korisnik mora imati vlastiti CPU, tada, prema sadašnjim saznanjima, neophodno moramo komunikaciju Sezam <-> korisnik obavljati po fizičkoj liniji Drugim riječima moral će se nabaviti uređaj koji multipleksiranu X 25 komunikacijsku liniju „raspreda" u određeni broj fizičkih linija Takav uređaj postoji i zove se PAD (tzv Packet Assembler Disassembler) Štoviše postoje i domaći isporučioci pa i proizvođači takvih uređaja te bi u suradnji s proizvođačem PAD a bilo moguće sasvim lijepo riješiti pitanja oko povezivanja Sezama na JUPAK

No PAD linije se ne ponašaju kao obični modemi a i signalizacija je nešto drugačija, pa bi i glavnu Sezamovu aplikaciju valjalo malo preprevititi te dopisati kontrolni program za PAD Krajnji bi rezultat bio sasvim upotrebljiv Sezam na JUPAK-u Doduše postojalo bi fizičko ograničenje budući da na jednom PAD-u ima od 6 fizičkih linija što u usporedbi s prosječno mogućih 128 virtualnih kanala po jednoj X 25 liniji nije mnogo, no kombinacijom više PAD-ova i SWITCH-eva je moguće broj fizičkih linija podići na sasvim pristojan broj i u ovakvoj koncepciji SWITCH je, naravno, takav uređaj koji jednu X.25 liniju „raspreda" u tri takve linije te tako omogućava spajanje više X.25 uređaja recimo PAD-ova u kaskadi

ZAKLJUČAK

Navedena rješenja su na sijaset instalacija u svijetu provjerena i potvrđena u praksi Ipak svaka mrežna instalacija ima svojih specifičnosti, te ne postoji jedinstveno pravilo o tome koje je rješenje bolje Tim više kada je glavni zadatak neke instalacije vezan uz jedinstvenu aplikaciju, kakav je recimo Sezam gdje aplikaciju pozna relativno uzak krug ljudi Tada se obično od krajnjeg rezultata dolazi suradnjom šireg kruga ljudi od kojih svatko dobro poznaje svoje područje, operativni sistem, glavnu aplikaciju i mrežu

SEZAM

BILTEN

Pripremа: Dejan Ristanović

5

MAJ 1991

Maj je na Sezamu protekao u znaku povećanog priliva mahom veoma atraktivnog PD softvera i, naročito, opisima raznih nedokumentovanih i slabo dokumentovanih opcija MS DOS-a i instrukcija Intelovih procesora. A što se konferencijskih događaja tiče, „obračun" u FORUM/devojke i igra *Prince of Persia* predstavljaju nešto što do sada nismo doživeli.

Kada je reč o bilansu, tokom aprila smo „opslužili" 13 235 poziva, komunicirali sa korisnicima čitavih 3 110 časova tj. 130 dana (obzirom da mesec ima 31 dan, u proseku je bilo neprekidno zauzeto 4,2 noda – da jeseni nam definitivno trebaju novi linije) od čega je u chatu provedeno 1 100 časova ili 137 radnih dana (i radnih noći). Korisnici su sa Sezama *download*-ovali 280 megabajta datoteka, mada je za isto vreme sistemu pristalo 18,5 megabajta.

Što se stvari „koje ostaju" tiče, tokom aprila je u konferencije pristiglo oko 2600 poruka (3 megabajta teksta), 3,7 megabajta datoteka priključenih uz njih, 4600 privatnih poruka (oko 2 megabajta privatne pošte, ne računajući datoteke uz te poruke), 2200 poruka u prošlog meseca uvedenim grupama (preko megabajt teksta) i za sta mnogo novih datoteka u direktorijumima.

Kada smo već kod datoteka, da kažemo da tom „grani" Sezama posvećujemo sve više pažnje. Naša koncepcija je od samog početka davala prioritet konferencijama nad prenosom datoteka, ali rast Sezama pokazuje da su same konferencije (i datoteke uz poruke u njima) „preuske" za potrebe naših korisnika i da se ilustracija pojedinih stvari o kojima se u stručnim konferencijama diskutuje meri stotinama kilobajta, što je mnogo preglednije i lakše za docnije nalaženje kada ga upišemo u direktorijum. Trudimo se, međutim, da ne prenosimo razne PC programe i slične kojih ima na hiljade, nego ekskluzivne materijale sa stranih sistema, kao što je bila dokumentacija za sve „ilegalne" interapte MS-DOS-a, tekstovi o kriptografskim algoritmima i tome slično. Tokom maja otvorili smo i direktorijum UNIX a u (srednjoročnom) planu je i otvaranje konferencije koja bi se bavila Unix-om i drugim ne-DOS operativnim sistemima na PC-ju (za sada se o tome diskutuje u posebnim temama konferencije PC SOFT). Najatraktivnije datoteke (najveći broj *download*-a) bile su neizbežni SCANV77.ZIP, pomenuta dokumentacija PC interapt sistema, testiranje brzine računara u MIPS-ovima, program VPIC42 za konverziju različitih grafičkih formata, novo rešenje YU slova za VGA karticu, *Telix 3.15*, par primera iz „Računara" i tako dalje.

Porastu broja datoteka u direktorijumima posvetio je i imenovanje moderatora tog segmenta sistema – to je *Jovan Bulajić (bulaje* na Sezamu).

U konferenciji **NOVOSTI** su tokom maja, kao i ranijih meseci, neprekidno pristizale „vruće" kompjuterske vesti sa BIX-a; preneli smo ih 119 odnosno oko 2,7 megabajta teksta. Ovde ćemo samo pomenuti izveštaje sa nekoliko američkih sajmova, nove domene u paketnoj mreži, sudske sporove oko *Windows* korisničkog interfejsa i novi *Hewlett-Packard*-ov laserski štampač.

Što se konferencije **RAČUNARI** tiče, imali smo „očekivane" velike debate o novom izgledu i sadržaju „Računara" koji je „bemelja" u aprilu, nešto diskusije o „Računarima 72" (prilično povoljno ocenjeni!), predloge za osnivanje SIG-ova (*special interest group*) koji bi se bavili pojedinim specijalizovanim temama (uglavnom profesionalnog) računarstva i koji bi bili objavljivani najpre na Sezamu a docnije i u „Računarima". No, kao što često biva u takvim slučajevima, pokazalo se da interesovanje bitno prevazilazi realne mogućnosti, tj. da ima mnogo onih koji bi rado čitali ovakve sadržaje ali malo onih koji bi ih pripremali.

Sezam je opsežno „pokrivao" majski Sajam tehnike u Beogradu („Tehnotronika 1991") kojim povodom je otvorena specijalna konferencija **SAJAM**. U okviru nje su naši saradnici detaljno izveštavali o kompjuterskim događanjima na Sajmu, prenosili cenovnike i ponude i tome slično dok su zainteresovani (ne samo naši pretplatnici nego i posetioci Sezamovog štanda kao i oni koji su telefonom zvali i koristili javno korisničko ime *sajam*) mogli da komentarišu te vesti kao i druge događaje na Sajmu. Pošto je manifestacija završena, konferencija je zatvorena za upis ali ostaje na sistemu kao demonstracija onoga što Sezam može da uradi – verovatno ćemo je ponovo aktivirati u povodu narednih sličnih manifestacija.

Obzirom da tokom maja nije bilo bitnih izmena u Sezamovom softveru, i konferencija **SEZAM** je bila relativno mirna. Bilo je nekoliko zanimljivih predloga o poboljšanjima *group* režima, diskusije o otvaranju novih teme i novih konferencija, mehanizma slanja datoteka uz poruke i u direktorijume, predloga za statistiku Sezama... Većina poruka se, naravno, odnosi na pitanje novih korisnika i odgovore moderatora.

I **KOMUNIKACIJE** su bile mirnije nego tokom aprila – pošto je diskusija o „najboljem" komunikacionom programu uglavnom završena bez zaključka (kao i uvek u takvim prilikama, svako nastavlja da koristi ono što mu se najviše dopada), vratili smo se „ovim" temama: fax karticama, problemima sa *zoom*-ovim, programima za kontrolu „udaljenih" PC-ja (dobili smo, na primer, *TeleReplica 2.0*)... „Šlag" meseca je bila uporedna analiza arhivera (PKARC, PKZIP, ZOO, LHA i ARJ) – pokazuje se da je ARJ najbolji ali razlike nisu drastične pa većina nastavlja da koristi PKZIP. Diskutovali smo, naravno, i o elektronskoj pošti, VAX-ovima (znate li kako da napravite svoj HELP na VMS-u?), problemima sa slanjem pošte u Ameriku, određivanju adresa korisnika čije ime znamo i tako dalje.

U konferenciji PC HARD stanje redovno – pošto se stalno postavljalo pitanje o raznim karakteristikama pojedinih diskova (broj glava/cilindara itd), uspeli smo da nabavimo jednu zaista gigantsku tabelu (100-tinak kilobajta ZIP-a) u kojoj su date karakteristike ne samo hard nego i laserskih diskova i disk kontrolera svih mogućih proizvođača – do sada nismo naišli ni na jedan model koji u toj tabeli nije opisan (tabelu je na BIX-u pronašao i nama preneo *Pavle Pekovic (ppekovic)*). Diskutovali smo i o raznim kvarovima na PC-ju, čišćenju disk jedinica, tipovima kućišta i tastatura i sličnim temama.

Što se konferencije PC SOFT tiče, ponovo smo diskutovali o softverskom isključivanju *NumLock* indikatora (problem je rešiv ali ne baš tako jednostavno kao što se na prvi pogled čini), tražili mogućnost da se otkrije da li je ANSI.SYS (ili neki ekvivalentan drajver) instaliran, pojavili su se prva iskustva sa novim *Borland C++* kompajlerom... Posebno su značajna upozorenja o „trojancima" koji se šire u okviru legalnih verzija SCANV-a i PKZIP-a – pokazuje se da je „lažna" registrovanih kopija PKZIP-a otkrivena i da nepoznati hakeri umeju da falsifikuju „signaturu" arhive pa je tako neko uložio veliki trud da falsifikovani SCANV liči na pravi. Bilo je mnogo poruka i u temi *virus* a počeo je da pristiže i PD softver, naravno u izvornom kodu.

Tokom proteklog meseca u konferenciji ATARI ST bilo je veoma promeltno, o čemu govori i lista poruka koja se približava broju 3000. Pričalo se i diskutovalo o programiranju (pisanju *accessory* programa), novom uslužnom softveru, a bilo je i dosta reči o najnovijem softverskom hitu, igri OXYD. Tema *news* je, kao i obično, obilovala prilozima među kojima su elektronski časopisi *STReport* i *Z-NET*, dok je u ATARIST direktorijum dospeo veliki broj programa, uglavnom uslužnih za rad sa hard diskom.

Neobično hladan i kišovit april „prikovao" je mnoge korisnike Sezama uz kompjuter pa su i u konferenciji ORKA glavna tema bile igre. Bilo je najviše saveta i trikova za hit igru *Prince of Persia* koja je dominirala i proteklim Sajmom tehnike; problem je jedino što oni (ne baš malobrojni) koji su rešili sve zamke ove igre (ili otkucali NU PRINCE SAV) i stigli do princeze nisu baš oduševljeni iskazima njena zahvalnost. Vlasnici HP48SX kao da su rešili da zasluže posebnu konferenciju – pored nekoliko igara (npr. novi Tetris) stigao je i SPICE (program za elektroniku), opis procesora sa naredbama itd. Zabeležena je zajednička prezentacija *Lotusa* i *Oracle*-a i odgovoreno na više pitanja, između ostalih, kako napraviti kvalitetnu fotografiju slike sa ekrana.

Maj u FORUM-u se, začudo, može okarakterisati opadanjem broja poruka na političke teme – razlog možda treba tražiti u tome što je nezanemarljiv broj korisnika početkom meseca stigao samo da ostavi poruku „stigao poziv" i da nestane u (u početku) nepoznatom pravcu. Čitavo to stanje je, naravno, „porkiveno" pratećom raspravom u konferenciji. Kada se situacija malo smirila, diskusija se „preselila" u temu *devojke*, ali se to pretvorilo u novokomponovani hit „prokleta je žena taaaa" u nastavcima i sa dosta neukusa – moderator i Uprava su imali prilično muke da stvari vrate u normalne tokove bez pribegavanja „drastičnim" sredstvima.

MODEMI SA PRETPLATOM

Beogradska firma ACS nudi veliki izbor modema po pristupačnim cenama. Pretplata na Sezam je uračunata u cenu, a sve formalnosti oko učlanjenja kupci ACS modema rešavaju već prilikom preuzimanja uređaja. Za detaljnija obaveštenja obratite se na adresu: **ACS Beograd**, S.A.M. Konsalting Centar, Ho Ši Minova 27, 11000 Novi Beograd, tel. (011) 136-322 i (011) 136-604, fax 136-32.

SEZAM BILTEN : MAJ [illegible]				
Konferencija	Poruka	Dužina	Datoteka	Dužina
UPUTSTVO	[illegible]	[illegible]	[illegible]	[illegible]
NOVOSTI	[illegible]	[illegible]	[illegible]	[illegible]
SAJAM	[illegible]	[illegible]	[illegible]	[illegible]
RAČUNARI	[illegible]	[illegible]	[illegible]	[illegible]
FORUM	[illegible]	[illegible]	[illegible]	[illegible]
KOMUNIKACIJE	[illegible]	[illegible]	[illegible]	[illegible]
PC SOFT	[illegible]	[illegible]	[illegible]	[illegible]
PC HARD	[illegible]	[illegible]	[illegible]	[illegible]
ATARI ST	[illegible]	[illegible]	[illegible]	[illegible]
ORKA	[illegible]	[illegible]	[illegible]	[illegible]
SEZAM	[illegible]	[illegible]	[illegible]	[illegible]
EXTRA	[illegible]	[illegible]	[illegible]	[illegible]
VICEVI	[illegible]	[illegible]	[illegible]	[illegible]
MALI OGLASI	[illegible]	[illegible]	[illegible]	[illegible]
Ukupno:	[illegible]	[illegible]	[illegible]	[illegible]
Privatne poruke:	[illegible]	[illegible]		
Grupe:	[illegible]	[illegible]		
Broj poziva				[illegible]
On-line vreme				[illegible]
Chat vreme				[illegible]
File transfer vreme				[illegible]
Upload datoteka				[illegible]
Upload Kb				[illegible]
Download datoteka				[illegible]
Download Kb				[illegible]

Maj je je bio vrlo buran i u konferenciji **VICEVI**. Najveću pažnju je izazvala novootvorena tema *hjak*, namenjena za mršavljenje (u nju stižu vicevi posle kojih čovek tri dana ne može ništa da jede). Izbor za vic meseca ovoga puta nije bio neizvesan pošto je od prvog trenutka „vodio" vic koji je najzad i pobedio sa do sada nezapamćenom većinom glasova (svi se bune što vic meseca ne prenosimo ovde, pa da ovoga puta probamo: „Mama, mama, kako to da sam ja crn a ti bela?" „Ćuti, sine, kakva je to žurka bila, srećan si što ne laješ"). Novost je da od sada nagrađeni može da bira – povećanje vremena, ili produženje pretplate za 30 dana.

KANAL ZA SARADNIKE

Ako želite da se preko „Sezama" obratite glavnom uredniku „Računara" sa ponudom za saradnju, a niste naš pretplatnik, to možete učiniti preko fiktivnog korisnika SARADNIK.

Na prompt **Username** odgovorite SARADNIK. Umesto lozinke, sistem će zatražiti da unesete svoje ime i prezime i grad iz koga se javljate. Nakon toga dobijate meni koji vam omogućuje da ostavite poruku za glavnog urednika, da uz tu poruku prikačite datoteku – rukopis koji nudite za objavljivanje, da proverite da li je sve prošlo OK i da napustite sistem.

Uz poruku obavezno ostavite svoje lične podatke. Pošto ne postoji tehnička mogućnost da vam bude odgovoreno preko Sezama, jer bi taj odgovor video svako ko se prijavi kao SARADNIK, to je jedini način da redakcija stupi u kontakt sa vama.

KAKO POSTATI ČLAN

Postupak učlanjenja počinje zvanjem Sezama. Nakon poziva broja 648-888 i uspostavljanja veze (parametri 2400N8) i pozdravne maske, dobićete pitanje

Username

Na njega treba odgovoriti sa **NEW** (novi korisnik), nakon čega vas Sezam vodi kroz proces prijave. Prvi podatak koji treba da date je vaše puno ime i prezime a potom birate pseudonim pod kojim ćete koristiti sistem. Pseudonim je bilo kakva reč do 15 znakova koja sadrži samo mala slova i znak . (tačka) i obično se bira tako da asocira na vaše ime. Sistem će na osnovu vašeg pravog imena predložiti pseudonim koji možete ali i ne morate prihvatiti. Pre izbora pseudonima dobro razmislite, jer on postaje vaše trajno ime na Sezamu i ne možete ga više promeniti.

Sledeći korak je izbor lozinke (password) kojom se vaš pseudonim štiti – prijavljivanje sistemu je moguće samo ako poznajete lozinku koja je vezana uz vaš pseudonim.

Nakon ovih osnovnih podataka sledi izbor nekoliko radnih parametara (objašnjenje ćete dobiti na ekranu) a potom unos ostalih ličnih podataka – adrese, telefona i podataka o opremi koju koristite za zvanje Sezama. Ovim je prva faza gotova.

Posebno obratite pažnju da **adresu unesete korektno** – ispravna adresa je jedini način da Uprava ostvari dalji kontakt sa vama kako bi se do kraja obavio posao vašeg učlanjenja.

Sledećeg radnog dana će vam biti poslato pismo u kome ćete primati i popunjenu uplatnicu. Na vama je da izaberete period na koji se učlanjujete i na osnovu toga na uplatnicu upišete iznos pretplate:

za 3 meseca	**400 din**
za 6 meseci	**700 din**
za godinu dana	**1100 din**

Nakon izvršene uplate, fotokopiju potvrde o uplati pošaljite na naznačenu adresu Uprave Sezama bilo poštom ili (daleko bolje) faksom. Po prijemu ove potvrde Uprava će vam istog radnog dana odobriti pristup sistemu čime postaje punopravan član Sezama. Istog dana će na vašu adresu biti poslato i Uputstvo za korišćenje sistema.

Ukoliko pri prvom prijavljivanju imate problema i ne završite popunjavanje podataka korektno nije potrebno da zovete ponovo i prijavljujete se pod novim imenom. Sačekajte jedan dan pa pozovite ponovo – administratori sistema koji detektovanje nepotpunih prijava intervenišu i daju vam mogućnost da pri sledećem javljanju ponovo popunite prijavu ili vam ostavljaju ličnu poruku kojom vas obaveštavaju šta treba dalje da učinite.

Ukoliko ste se ipak prijavili pod više različitih pseudonima, imajte na umu da Uprava sama može da izvrši izbor koju od vaših prijava će uzeti u dalju obradu. Kriterijum je najčešće kompletnost podataka. U slučaju više korektnih prijava uzima se u obzir samo poslednja. U svakom slučaju, na uplatnici će biti upisan pseudonim na koji se uplata odnosi i samo njega možete koristiti.

Ukoliko želite pre prijavljivanja da razgledate sistem, na pitanje Username odgovorite sa DEMO. Naći ćete se u demonstracionom režimu, radu u kome možete pregledati kompletnu strukturu sistema koja je identična onoj „pravoj". Broj DEMO poziva nije ograničen a vreme za jednu seansu je 5 minuta.

NA SOPSTVENI RIZIK

Verovatno sa povremeno sa nostalgijom setita dana kada smo do detalja rastavijali ROM-ove naših računara, poznavali svaku siatemaku promenjivu i izvodili gotovo nemoguće trikove – pretvarali blok grafiku u grafiku visoke razolucije, ubrazavali čitanja sa kaaete ili diska deset puta u odnosu na ono što je firma predvidela ili pravili „yu bejzik". Pojavom DOS-a koji ja konačno na naše atolova stavio nešto što liči na pravi operativni sistem, a pogotovo pojavom UNIX-a ili OS/2 na PC mašinama, draž ja nastala: sva što traba uraditi je dokumentovano, a ono što nija – to je zabranjeno!

Zoran Životić

Ipak draž „hakerisanja" nije do kraja nestala sa DOS-om. Iako je okolnost da je u upotrebi istovremeno više verzija DOS-a (danas su rasprostranjene verzije 3.30, 3.31, 4.0 i 5.0) i to od različitih firmi (IBM, Microsoft, COMPAQ i neke druge) potpuno onemogućila da se koriste najniže moguće tačke operativnog sistema, ipak jedna oblast – nedokumentovani sistemski pozivi – omogućava kreiranje interesantnih i prilično pouzdanih programa. Pri tom se, zapravo, ne radi o nečem što je potpuno nelegalno sa stanovišta operativnog sistema – sam DOS i njegove naredbe obilno koriste ove pozive. Ilustrovaćemo to na jednom primeru.

PROGRAMI IZ POZADINE

Što se Microsofta i odnosa sa programerskom javnošću tiče, ne postoji legalan način da se napiše program koji bi, pod tajmer interaptom, čitao neku datoteku i u malim porcijama slao ispis na štampač. Realizacija ovog programa bi zahtevala da se teorijski u bilo kom trenutku (tajmer interapt je hardverski i dešava se potpuno nezavisno od stanja operativnog sistema) izvede program koji otvara datoteku, čita ili zapisuje u nju i šalje na štampač niz znakova. Zbog poznate i više puta spominjane osobine da DOS ne dozvoljava rekurzivne pozive, te da nije moguće DOS poziv koji, recimo, čita datoteku prekinuti i pozvati tu istu DOS funkciju deluje sasvim nemoguće napisati takav program. Pa ipak, on postoji – napravio ga je sam Microsoft u obliku PRINT naredbe koja radi upravo opisan posao. Način na koji je taj program napisan nije ni malo nelegalan – naprotiv, radi se o korišćenju isključivo postojećih DOS funkcija na sasvim uobičajen način – jedino što te funkcije nisu javno dokumentovane. Očigledan razlog zbog koga ovo nije učinjeno je što bi time firma imala obavezu da održava kompatibilnost među verzijama, i to u oblasti koja je van osnovne koncepcije sistema. Sa PRINT nardbom ne postoji problem kompatibilnosti, jer se ona uvek uparuje sa samim operativnim sistemom.

Ipak, programi koji „iskaču iz pozadine" u sred drugih programa su izuzetno atraktivni, pa iskušenju nije odolela ni tako velika firma kao što je Borland. *SideKick* je jedan od prvih komercijalnih programa koji se kockao sa mogućnošću da ga neka sledeće verzija DOS potpuno izbaci iz upotrebe zbog nekompatibilnosti. Ne znamo sa kojim činjenicama je računao Borland kada se upustio u ovakav projekat. Tek, pokazalo se da nije pogrešio – nedokumentovane funkcije na kojima se bazira *SideKick* su i do danas ostale iste.

Moram priznati da mi je do skoro posao pisanja ovakvih programa delovao krajnje odbojno, jer je po prirodi stvari, sve upućivalo da tu treba koristi i asembler i snalaziti se na razne načine kako bi se stiglo do pouzdanog rada. Kada sam već bio primoran da se ovim problemom pozabavim, pokazalo se da je rešenje prilično zabavno i da se dosta brzo, uz pomoć isključivo C-a, može doći do raznih korisnih programa. Evo kako je to izgledalo.

Zadatak je bio napraviti program koji će u pozadini kontrolisati komunikacioni port i sve što preko njega stiže zapisivati u datoteku. Primena ovakve alatke je dosta široka – od „hvatanja" ispisa bilo kog programa u datoteku (umesto com porta može se preuzeti BIOS funkcija za ispis na štampač), do pravljenja kompletnog komunikacionog programa koji bi u pozadini pozvao neki servis i obavio isprogramiran zadatak. Kao što ću na kraju pokazati, primer se uz neznatnu modifikaciju može pretvoriti u klasičan program *SideKick* tipa koji na određenu kombinaciju tastera „iskače" iz pozadine i preuzima kontrolu nad računarem.

JEDNOSTAVNA ZAMISAO...

Dakle, klasičan komunikacioni deo koda (o ovome je bilo reči u jednom od ranijih brojeva, vidi *Računare 57, Bit po bit poruke*) pod interaptom prime znake i slaže ih u interni bafer – ostaje nam „samo" da ovaj sadržaj povremeno snimimo na disk – sve to, naravno, u pozadini, ne ometajući računar u njegovom tekućem poslu. Prvo što se nameće kao pitanje je – može li se ovakav program realizovati potpuno u C-u? Odgovor je – može. Pošto se ovde nećemo baviti razradom konkretnog zadatka, već nam je cilj da dođemo do osnova na koju svaki C programer lako može nadzidati bilo kakav svoj program, zanemarićemo činjenicu da se deo koji se bavi komunikacijom obično piše u asembleru. Pregled MS biblioteke C funkcija otkriva da postoji **_dos_keep** koji omogućava da bilo koji C program nakon završetka rada ostane rezidentan u memoriji. Sličnu funkciju imaju i svi ostali C kompajleri namenjeni DOS-u i PC-u. Takođe, deo biblioteke su i fukcije za preuzimanje interapt vektora (**_dos_getvect** i **_dos_setvect**), tako da već imamo dovoljno osnova za početak pisanja.

Prvo ćemo otvoriti jednu datoteku za zapisivanje, potom ćemo preuzeti interapt vektor 8 (tajmer) i napustiti program **_dos_keep** fukcijom. Dopisaćemo novu tajmer rutinu – unutar nje ćemo prvo pozvati originalni deo BIOS koda kako bi obavio svoj standardni posao, a potom proveriti stanje komunikacionog bafera – ako se nakupilo dovoljno znakova, zapisaćemo ih u datoteku koju smo pri startovanju otvorili. Gore opisano može da stane u recimo trideset linija C koda, ali će vas startovanje ovog programa brzo uveriti da je pravljenje bekapa celog diska prva operacija koju treba uraditi kada se krene sa ovakvim poslom! Uz malo sreće može vam se desiti da se računar samo zaglavi – verovatnije je da će FAT tabela diska doživeti nepopravljiv lom. Razlog je jasno zapisivanje na disk – tajmer prekid ima dosta šansi da uhvati neku drugu DOS funkciju u sred njenog izvršavanja. Vreme je da se sastave i analiziraju informacije sa raznih strana o nedokumentovanim DOS fukcijama.

...I VELIKE KOMPLIKACIJE

DOS funkcije broj 34h vraća adresu internog DOS flega koji se naziva **Critical Section Flag** – njegova vrednost je 0 samo onda kada sistem nije unutar DOS koda, te je u tom trenutku moguće pozvati bilo koju DOS funkciju. Lako možemo modifikovati program tako da u tajmer rutini, pre nego što krene da zapisuje na disk, proveri stanje ove zastavice – ako nije 0, prosto možemo preskočiti taj ciklus i sačekati neko bolje vreme. Startujemo modifikovani program i... ništa. Tajmer rutina će stalno dobijati da je fleg različit od nule i da ne sme da koristi DOS. I taman kada pozovete editor da pogledate gde ste to pogrešili u programu – ispis na disk će „provaliti" svom silinom! Malo razmišljanja će ukazati na očiglednu činjenicu – dok se unosi komandna linija, sistem se nalazi unutar DOS funkcije broj 0Ah i nije ni čudo što dalji DOS pozivi nisu dozvoljeni!

Ovakvo stanje nas dovodi u pomalo smešnu situaciju – naš program će da radi samo onda kada računar radi nešto drugo – onda kada ima najviše vremena i čeka da zadamo komandu, sprečen je da obavlja posao. Na sreću, autori operativnog sistema su se podnnuli za ovaj detalj. Sve vreme dok čeka na taster DOS periodično izvodi instrukciju INT 28. Poziv ovog interapta je tako pripremljen da omogućava da se unutar rutine za njegovu obradu takođe izvede bilo koji DOS poziv. INT 28 se naziva *Idle Interrupt* („besposlen sam" interapt) i zapravo je i smišljen zbog ovakve vrste programa.

Dakle, novi dodatak programu – preuzećemo interapt vektor 28h i pozvati istu rutinu za proveru stanja komunikacionog bafera – ovaj put pre zapisivanje na disk ne treba da proveravamo stanje *Critical Section* flega. Startujemo program i... ništa. Program će korektno raditi, ali će svaki pokušaj zapisivanja na disk javljati grešku.

Otkrivanje grešaka u programima ovog tipa je veoma zanovetan posao – *CodeView* i slične alatke nisu od velike pomoći, pa se obično treba snalaziti ubacivanjem dodatnog koda koji ispisuje nastale greške. Pokušaj da u ovom programu ubacimo jedan **printf** nakon poziva funkcije za ispis u datoteku da vidimo o kom kodu greške se radi će nas dovesti do spoznaje da ima još problema koje treba rešiti. **Printf** će korektno ispisati rezultat kada smo na nivou DOS prompta – kada startujemo neki program, **printf** iz pozadine će zaglaviti računar što ukazuje na neko „brljanje" po memoriji. Ipak, ovaj problem možemo ostaviti za kraj.

Poznato je da DOS čuva negde informaciju o datotekama koje je program otvara – prekid rada programa će zatvoriti sve datoteke, čak i

Listing 1 tsr main()

```
/*
** OSNOVA TRS PROGRAMA
** Racunari 73
** Autor Zoran Zivotic
*/
#include <dos.h>
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <share.h>
#include <io.h>
#include <stdarg.h>
#include <memory.h>
#include <string.h>

#pragma check_stack( off )
#pragma check_pointer( off )

void (_interrupt _far *oldtimer)( void );
void (_interrupt _far *oldidle)( void );
void (_interrupt _far *oldkey)( void );
void _interrupt _far newtimer( void );
void _interrupt _far newidle( void );
void _interrupt _far newkey( void );
void do_tsr(void);
void tsr_main(void);
void ctrl_c(void);

char _huge *programend;
char _huge *programbeg;
char _far *savestack;
char _far *idleflag;
int i_am_busy = 0;
int jump_in = 0;
unsigned savepsp,
char _far *kb_flag = (char _far *)0x4171;


void main( int argc, char *argv[] )
{
    unsigned ressize;

    _asm {
        mov  word ptr programend[0], sp
        mov  word ptr programend[2], ss
        mov  ax,0x3400
        int  0x21
        mov  word ptr idleflag[0], bx
        mov  word ptr idleflag[2], es
    }
    FP_SEG( programbeg ) = _psp;
    FP_OFF( programbeg ) = 0;

    ressize = ( (programend - programbeg) >> 4 ) + 64;

    oldtimer = _dos_getvect( 0x08 );
    _dos_setvect( 0x08, newtimer );

    oldidle = _dos_getvect( 0x28 );
    _dos_setvect( 0x28, newidle );

    oldkey = _dos_getvect( 0x09 );
    _dos_setvect( 0x09, newkey );

    _dos_keep( 0, ressize );
}

void _interrupt _far newtimer(void)
{
    (*oldtimer)();
    if ( jump_in && i_am_busy == 0 && *idleflag == 0 )
        do_tsr();
}
void _interrupt _far newidle(void)
{
    (*oldidle)();
    if ( jump_in && i_am_busy == 0 )
        do_tsr();
}
void _interrupt _far newkey( void )
{
    (*oldkey)();
    if ( i_am_busy == 0 && jump_in == 0)
        if ( ( *kb_flag & 6 ) == 6 )  /* shift + ctrl */
            jump_in = 1;
}

void do_tsr()
{
    i_am_busy = 1;
    _asm
    {
        cli
        mov  word ptr savestack[0],sp
        mov  word ptr savestack[2],ss
        mov  sp,word ptr programend[0]
        mov  ss,word ptr programend[2]
        sti

        mov ah,0x51
        int 0x21
        mov savepsp,bx
        mov bx, _psp
        mov ah,0x50
        int 0x21
    }

    tsr_main();

    _asm {
        mov bx,savepsp
        mov ah,0x50
        int 0x21
        cli
        mov  sp,word ptr savestack[0]
        mov  ss,word ptr savestack[2]
        sti
    }
    i_am_busy = jump_in = 0;
}

void tsr_main()
{
    int c = 7;

    _dos_write(1,&c,1,&c);
}
```

Univerzalna osnova rezidentnog programa

kada to sami ne uradimo. Dokumentacija nas uverava da DOS funkcija 31h (**Terminate & stay residnet**) ne zatvara otvorene datoteke, što važi i za C funkciju **_dos_keep** koja se na ovu oslanja. Dakle, datoteka jeste otvorena ali je DOS nekako „pogubio" informaciju o njoj. Vreme je za nove nedokumentovane DOS pozive.

no, zapisivanje na disk – tajmer prekid ima dosta šansi da uhvati neku drugu DOS funkciju u sred njenog izvršavanja. Vreme je da se sastave i analiziraju informacije sa raznih strana o nedokumentovanim DOS fukcijama

Deo informacije o otvorenim datotekama se čuva u PSP oblasti (*Program Segment Prefix*) – radi se o bloku od 256 bajtova koji DOS automatski kreira pre startovanja svakog programa – u toj oblasti se nalazi i inicijalni stek prostor, kao i inicijalno DTA (*Disk Transfer Area*) područje. Sistem u svakom trenutku drži jednu takvu oblast aktivnom – kada se startuje program, aktivan je njegov PSP i sve informacije o datotekama se pamte na tom mestu. Ono što se desi sa našim programom je da po prekidu njegovog rada i startovanju nekog drugog programa njegov PSP ostaje zaboravljen.

Naravno, postoji rešenje i za ovaj problem – ono se krije u dve nedokumentovane DOS funkcije broj 50h i 51h. Druga će nam vratiti adresu trenutno aktivnog PSP-a – nju ćemo sačuvati, potom funkcijom 50h proglasiti naš PSP aktivnim i nakon obavljenog posla vratiti stanje nazad. Nakon ove intervencije program će već početi sasvim „razumno" da se ponaša.

NAZAD KAKO TREBA

Ostaje nam da razrešimo još i pitanje zašto upotreba **printf** funkcije pravi haos po memoriji. Odgovor će se nametnuti još na samom početku pisanja programa kada budemo ozbiljnije odlučili da analiziramo parametre **_dos_keep** funkcije. Prvi parametar je kod greške koji treba vratiti DOS-u (ako je sve u redu, vraćamo 0) a drugi parametara je broj paragrafa memorije (paragraf - 16 bajtova) koji treba rezervisati za naš program u rezidentnom stanju. Koliko to memorije treba našem C programu?

Odgovor na ovo pitanje je prilično delikatan

ako malo ne ograničimo naš zahtev. Rezidentan program i ovako ne bi trebao da zauzima previše memorije, pa nije neki problem ograničiti se na **small** memorijski model (64K kod + 64K podaci). U tom slučaju situacija je relativno jasna – početak programa je uvek PSP područje (segment ove adrese je definisan kao globalna varijabla **_psp**), a kraj je stek segment. Malom manipulacijom sa **_asm** direktivom možemo vrednost stek segmenta prebaciti u neku promenjivu i potom oduzimanjem od vrednosti **_psp** promenjive stići do potrebne količine memorije.

Na žalost, ovakav račun ujedno znači da potpuno otkidamo **heap** (prostor za dinamičku alokaciju) koji se nalazi iza steka. I to, izgleda, na način koji ne stavlja do znanja **malloc** funkciji da smo joj potpuno ukinuli mogućnost za rad. Zato će svaki poziv bilo koje funkcije koja interno poziva **malloc** (a **printf** to čini) jednostavno krenuti da piše po tuđem delu memorije što more imati loše posledice.

Rešenje ovog problema verovatno postoji, ali bi zahtevalo dosta petljanja – možda se odgovor krije u statičkom deklarisanju niza **heap**, pa potom njegovog uključivanja u **heap** prostor funkcijom **_heapadd**. Ipak, ovakvo rešenje poznaje samo MSC 6.0 pa se nismo upuštali u njegovu preveru. Zadovoljili smo se ograničenjem da ne koristimo originalni **printf** (lako se može dopisati ekvivalent zahvaljujuću **vsprintf** funkciji), kao i da umesto **fopen/fread** itd funkcija koristimo **_dos_open** i **_dos_read** (ili **open/read**) pogotovo što će nam biti potreban *file share*-ing koje funkcije **f** klase ne poznaju.

Program je sada konačno profunkcionisano kako treba. Ono što smo uradili bez mnogo analize je da u glavnoj funkciji odmah prebacimo stek na naš originalni – stanje steka je verovatno u svakom trenutku zadovoljavajuće veliko da omogući poziv nekoliko funkcija, ali nema potrebe da rizikujemo – originalni stek programa je sigurno dovoljno velik.

Zadovoljstvo što program radi kako treba pokvarila je još jedna neoprezno učinjena koncepcijska omaška. Naime, podaci koje zapisujemo u datoteku mogu se pregledati i u toku izvršavanja programa. Predvideli smo, zbog mogućeg rada pod mrežom, da osnovni program otvara datoteku sa pravom čitanja/pisanja, a da zabranjuje ostalima pisanje. Ovo je, ujedno, značilo da treba napisati novi programčić koji je ekvivalent TYPE naredbe, ali umesto ekskluzivnog prava na datoteku koje TYPE traži, ostvara je samo za čitanje i bez zabrane ostalima. Za vreme dok je program aktivan, u datoteku pristižu podaci – našom TYPE komandom možemo pregledati u bilo kom trenutku njihov sadržaj, iako će dužina datoteke u direktorijumu stalno biti prikazivana kao 0 bajtova, jer DOS ovaj podatak upisuje pri zatvaranju datoteke.

Sve ovo bi bilo u redu da nije jednog sitnog problema – program nigde ne predviđa zatvaranje datoteke! Kada ugasimo računar na disku ćemo samo ostaviti *lost clusters* i sadržaj datoteke će biti praktično izgubljen. Inicijalno otvaranje datoteke zato treba preskočiti, pa svako zapisivanje pretvoriti u ciklus open/write/close, ili programom predviđati zatvaranje datoteke nekom posebnom procedurom, što je nešto teži put.

UNIVERZALNA TSR OSNOVA

Sada već imamo jasan kostur bilo kakvog rezidentnog programa koji može da radi i nešto više od štampanja sadržaja ekrana – tajmer interapt zajedno sa interaptom 28 će proveravati u kom trenutku program sme da preuzme kontrolu i da radi „šta mu je volja". Kada već imamo ovako zgodno rešenje, možemo dodati još jedan sitan korak i učiniti ga univerzalnim. Preuzećemo interapt broj 9 koji se generiše svaki put kada pritisnemo ili otpustimo taster. Naša rutina će prvo pozvati originalni BIOS kod da bi ovo stanje bilo korektno obrađeno, a potom možemo proveriti stanje flegova za **shift** tastere (nalazi se na adresi 417h). Ukoliko primetimo da stanje odgovara nekoj našoj izbaranoj kombinaciji – recimo levi **shift** + **control** možemo samo setovati neku našu promenjivu (u našem primeru **jump_in**). Tajmer i int 28 rutini potom ostavljamo da primete da je ovaj zahtev postavljen i da, kada su i ostali zahtevi zadovoljeni, startuju našu glavnu proceduru. Naravno, moguće je i odmah unutar same rutine koja zamanjuje int 9 proveriti da stanje DOS *Critical Section* flega i odmah pozvati glavnu proceduru. U tom slučaju možemo izbaciti INT 8, ali i dalje ostaje potreba za INT 28, tako da rešenje ne bi bilo mnogo kraće.

Tako smo konačno stigli do univerzalne osnove bilo kakvog TSR programa. Funcija **tsr_main()** ne obavlja skoro nikakav zadatak – ispisuje putem DOS-a karakter kod 7, što će proizvesti jedan beep – u nju možete ubaciti bilo šta poštujući naravno napomene koje smo dali o alokaciji memorije.

Na kraju, ostaje mi samo da dodam ono što rade i svi drugi koji se dotiču pitanja nedokumentovanih DOS funkcija – sve što budete radili sa podacima i kodom iz ovog teksta činite na sopstveni rizik! Da za ovakvo upozorenje ima osnove govori i činjenica da su informacije do kojih se može doći o nedokumentovanim DOS funkcijama često protivurečne i da u realizaciji programa različiti autori koriste neke nijanse za koje se ne vidi logično objašnjenje – verovatno se radi o njihovom iskustvu. Tako se i ovo što je pred vama može smatrati samo jednim iskustvom koje sigurno radi u kontekstu u kome je upotrebljeno. Ipak, ne treba da se iznenadite ako, šireći program, zapadnete u neku neočekivanu situaciju.

U PETOJ BRZINI

Nastavak sa strane 5

Performanse

Prilikom merenja posebnu pažnju smo obratili na performanse podsistema diska jer je to jedina EISA kartica u sistemu. Međutim, podsećamo da kontroler takođe sadrži i 512 KB keš memorije, pa je prilikom merenja bilo veoma teško razlučiti efekte sabirnice od efekata keš memorije. Performanse su dobre, čak izuzetno dobre ako se uzme u obzir da nismo potrošili ni jedan jedini bajt na spoljno keširanje diska, ali ipak nisu toliko dobre da briljiraju. Recimo, na testu sa bazama podataka, dobijen je indeks od 2.12.1, što nije sjajno. Poređenja radi, kada inače oho-ho sporiju R/Optimu 386 Mk II malo „potpomognemo" PC-Kwik-om, dobijamo indeks od 2.80 u mašinskim i čak 4,23 u programskim testovima. Jeste da je to sistemska memorija koja je naravno mnogo brža od one na kontroleru, ali je tačno i to da između takvog keša i procesora postoje čak tri nivoa programa, na kojima se ipak gubi vreme.

Što se opšteg nivoa preformansi tiče, nema sumnje ko je tu general a ko kapler. Procesor 80486, koji je zapravo jedan evolucioni korak napred od procesora 80386 (integrisaniji je nešto nalik razlici između 8086 i 80186), je jednostavno nedostižan. Uprkos ne baš sjajnim performansama kontrolera, ipak je uspeo da bude 3,31 puta brži od reference, 66% brži od R/Optima 386 Mk.II (bez Max-a i PC Kwik-a) i 19% brži od AMD ploče na 40 MHz. Ako isključimo disk, onda je 486 brži od reference za 3.9 puta, 67,7% od R/Optime i 18.2% od AMD ploče.

U programskim testovima, ukupni indeks preformansi je iznosio 3.02 puta u odnosu na referencu, rezultat koji je opet bio pod uticajem kontrolera, jer je računar „zenbao" (relativno gledano, u odnosu na svoju kompleksnost i cenu) na testovima sa diskom, a posebno sa bazom podataka, gde je zaredio indeks od ipak relativno malih 2.1. To je istina 70% brže od ploče sa procesorom AMD, ali podsećamo da je EISA 32-bitna sabirnica, pa bi samo po tom osnovu trebalo da bude dvostruko brža, zatim, tu je i veći radni takt, i najzad keš na kontroleru. Očekivali bismo više od takvog spoja.



Zapravo, ovde imamo jedan mali apsurd, kontroler da facto radi dva puta brže od standarda i dobrih 70% brže od inače brzog diska Quantum, a mi smo nezadovoljni, ipak smo očekivali više, ne desat puta više, ali brat-bratu 4-5 puta brže. Možda bi, sa većim kešom, to tako i bilo, ali isto bi važilo i za AT Bus kontroler sa sostvenim kešom, a da i ne govorimo o upotrebi sistemske memorije za keširanje.

Dakle, radi se o veoma moćnoj mašini sa naprednom tehnologijom i jednim delom u sebi koji je na žalost podbacio i uspeo da pokvari opštu sliku. Nema sumnje da je EISA standard budućnosti, pa na osnovu jednog, rekli bismo nedorečenog dela, ne treba osuditi čitavu tehnologiju.

* * *

Na kraju, ostaje nam da se opredelimo između ove dve međusobno različite mašine sa samo jednom zajedničkom crtom – performansama. Kome trebaju performansa bez ograničenja, neka se opredeli za platformu sa procesorom 80486, njegove brzine zadivljuje. Oni koji ipak razmišljaju o budžetu (uz melodiju razlike u ceni od oko DEM 2000), a kojima ne treba numerički koprocesor, neka se opredele za ploču sa AMD procesorom na 40 MHz, ovo posebno važi za ljude koji se bave DTP-om, jer je u opisanoj konfiguraciji AMD ploča dala indeks na video testovima za 20% bolji od ploče sa 80486 (odnosno, za 79,5% je brža od R/Optime 386 Mk II sa istom video kartom i 4.92 puta je brža od reference).

POSTANI WINDOWS PROGRAMER

Dosad smo se uglavnom bavili prikazima komercijalnih *Windows* programskih paketa. U ovoj seriji takstova pažnju uameravamo na drugu stranu medalja – razvoj aplikacija za rad u *Windows* okružanju.

Dejan Dinčić

Microsoft *Windows 3.0*, u kombinaciji se DOS-om trenutno je najaktuelniji PC operativni sistem na svetu. Od tranutka kada je najavljen, za njega se prerađuju i objavljuju najpoznatiji softvarski paketi iz DOS sveta. Iako *Windows* omogućava izvršavanje DOS programa, naročito na 386 računarima sa dovoljno memorije, ipak se tek sa programima koji su posebno pisani sa rad u ovom okružanju postiže pravi efekat. Ovo se odnosi kako na racionalnu upotrebu memorije i postignute perfomanse, tako i na velika grafičke potencijale. *Windows* postavlja donekle visoke zahtave pred hedrver računara na kome se može optimalno koristiti, ali nam je zbog stalnog pada cena i napretka tehnologije i takva oprema svakim danom sve bliža.

PROGRAMSKI ALATI

Konanici računare su zbrinuti dovoljnom količinom softvera koji radi pod *Windows*-ima. Kako stoji stvar se programerima i razvojem programa za *Windows*? Postoji nekoliko programskih alata za razvoj *Windows* programa. Standardni alat za razvoj *Windows* programa je *Microsoft* (čiji bi inače ?) *Windows 3.0 Software Development Kit* (SDK), uz upotrebu C kompajlera i asemblera iste firme. Nedavno se i Borland uključio u igru svojim novim paketima, od kojih su najveća novina *TurboPascal za Windows* i *Borlan C++*. Mi ćemo u ovoj seriji koristiti Microsoft alatka.

SDK sadrži biblioteke sa linkovanje sa objektnim kodom, dafinicija koje uz njih idu, i nakoliko pomoćnih programa. Pre svega, tu je *Resource compiler*, čija je namena da u program uključi i objekte koji nisu deo izvornog C koda, a koji su elementi *Windows* programa: menije, dijaloge, ikone i sl. Tu su još font editor, programi za kreiranje grafičkih alemanata aplikacije, i alati za dabagiranje. Ovom prilikom spomenućemo osobine ovog alata samo onoliko koliko je nužno za jasniju sliku o programiranju za *Windows*.

Osobina programa koje rade pod *Windows*-ima određene su specifičnostima ovog okruženja. Dominiraju dve osobina: multitasking i grafička okolina.

SISTEM PORUKA

Postupak deljenja procesora između više aktivnih programa razni operativni sistemi rešavaju različito. Procesi (delovi programa) koji se izvršavaju u *Windows* okruženju dobijaju kontrolu preko poruka koje im operativni sistem prosleđuje (*message driven system*). Sistem poruka je tehnika koju *Windows* konsti se realizaciju multitaskinga, ali istovremeno i način komunikacije programa sa operativnim sistemom i drugim programima koji se u tom trenutku izvršavaju. Sa stanovišta programa, sistem predaja poruku kao obaveštenje o nekom događaju: pomeren se miš, prozor smanjen, klik na neku opciju menija, taster pritisnut i slično. Broj poruka, i informacije u njima, su daleko obimnija od onog što prosečan program treba da obradi. Program zato reaguje samo na ona poruka koja utiču na njegovo izvršenja. Naš program takođe može da šalje poruka sistemu, ili drugim programima.

Treba imati na umu da, uz poruku, program od operativnog sistema dobija i kontrolu procesora. To znači da obradu poruka treba obaviti maksimalno brzo i vratiti procesor sistemu. Koncepciju programa treba prilagoditi ovoj tehnici i posao koji program obavlja razložiti na dovoljno malih celina. Izvršavanja ovih celina treba da se odvija kao reakcija na primljenu poruku i na sme trajati predugo, kako se ne bi narušio normalan rad sistema.

Druga bitna osobina koju nameće multitsking koncepcija zapravo je preporuka kako program treba da se ponaša u ovim uslovima. Program koji treba da radi pod *Windows*-ima mora voditi računa da nije sam na računaru i da *Windows* izvršava paralelno više procesa. Procesor računare je naizmenično na raspolaganju svakom od aktivnih procesa. Program se, dakle, izvršava u nekim vremenskim intervalima. Između dva uzastopna predavanja kontrola našem programu puno se toga može promeniti u sistemu: količina slobodne memorije, status datoteka i slično. Zato je potrebno da lepo vaspitan *Windows* program sve dinamične parametre proverava pred svako njihovo korišćenje. Direktno brijanje po memoriji i slične programerske poslastice treba zaboraviti. Većinu DOS poziva je moguće koristiti, ali sa povećanom pažnjom. *Windows* jednostavno nameće „čistije" programiranje, a usluga koje pruža nadoknaduju gubitak grubljih (pod DOS-om tako potrebnih) programerskih tehnika.

GRAFIČKO OKRUŽENJE

Windows pruža i grafičko okruženje. Jasno je da korišćenja uobičajenih funkcija sa ispis nije ni dovoljno ni moguće u novim uslovima. *Windows* sadrži veliki broj funkcija za ispis teksta, manipulaciju fontovima, kurzorima, ikonama, bitmapama, crtanje i popunjavanje površina. Složenost ovih funkcija veoma varira – u nekim slučajevima se jednim programskim redom obavija prilična količina posla, dok je za neka jednostavna rešenja potrebno ispisati desetak redova izvornog koda.

Uz *Windows* grafičko okruženja se vezuje još jedna, veoma važna prednost: hardverska nezavisnost programa. Pišući program ne morate voditi računa o uređajima na kojima će se program izvršavati. Gde god se *Windows* može instalirati, tamo će raditi i vaš program. Na ovaj način programer je oslobođen brige o raznim video adapterima, problemima kompatibilnosti različitih tipova štampača i druge oprame. Nezavisnost programa od hardvera ide i korak dalje: nezavisnost i od vrste izlaznog uređaja. Isti deo programa može da piše i crta na ekranu, štampaču i ploteru. Zapravo, program se ni u jednom trenutku na obraća direktno izlaznom uređaju, već crta po fiktivnom prostoru (*display context*). Deo operativnog sistema (*graphics device interface*) se stara da mapira i prilagodi konkretnom uređaju sve što program ispisuje. Jedino ograničenje ovoj pogodnosti je postojanja drajvera sa neki komad hardvera, ali s obzirom da se *Windows* nameće kao standard, danas je retko naći uređaj za koji ne postoji odgovarajući drajver.



Izbor opcije u meniju (program rac.c)



About dijalog aktiviran iz manija

Listing 1 rac.rc

```
NAME        Racunari

DESCRIPTION 'Simple Microsoft Windows Application'

EXETYPE     WINDOWS

STUB        'WINSTUB.EXE'

CODE        PRELOAD MOVEABLE DISCARDABLE
DATA        PRELOAD MOVEABLE MULTIPLE

HEAPSIZE    1020
STACKSIZE   5110

EXPORTS
        MainWndProc     @1
        About           @2
```

U risora datoteci rac.rc opisani su meni, akcija za odabrane tastere i dijelog sa programom

Listing 2 rac.def

```
/* opcije menija */

#define     IDM_EXIT     101
#define     IDM_ABOUT    102


/* Control IDs */

#define     IDC_BUTTON   401


/* funkcije    */

int PASCAL WinMain(HANDLE, HANDLE, LPSTR, int);
BOOL InitApplication(HANDLE);
BOOL InitInstance(HANDLE, int);
long FAR PASCAL MainWndProc(HWND, unsigned, WORD, LONG);
BOOL FAR PASCAL About(HWND  unsigned, WORD, LONG);
void RacunariPaint (HWND, HDC);
```

U datoteci rac.def moraju biti pobrojane sve funkcija programa koje direktno pozive operativni sistem

Listing 3 rac h

```
#include "windows h"
#include "rac.h"

RacunariMenu MENU
BEGIN
    POPUP        "&Options"
    BEGIN
        MENUITEM   "E&xit",              IDM_EXIT
        MENUITEM   SEPARATOR
        MENUITEM   "&About racunari...", IDM_ABOUT
    END
END

RacunariAcc ACCELERATORS
BEGIN
    0x81,       IDM_ABOUT,   VIRTKEY, ALT
END

AboutBox DIALOG 22, 17, 144, 75
STYLE WS_MODALFRAME | WS_CAPTION | WS_SYSMENU
CAPTION "About racunari"
BEGIN
    CTEXT "Racunari Application"  -1,       0, 10, 144, 8
    CTEXT "Version 1.0"           -1,       0, 25, 100, 8
    CTEXT "racunari 73/91"        -1,       0, 80, 144, 8
    DEFPUSHBUTTON "OK"          IDOK,      53, 59,  32, 10,      WS_ABOUT
END
```

Zaglavije programa

Listing 4 rac.c

```
/****************************************************************************

    PROGRAM: rac c
           ispisuje tekst "racunari"

    FUNKCIJE:
        WinMain() - uspostavlja petlju prometa porukama
        InitApplication() - registruje klasu prozora
        InitInstance() - kreira glavni prozor i botton
        MainWndProc() - prima i obradjuje poruke za glavni prozor
        About() - obradjuje poruke za "About" dialog box
        RacunariPaint() - Ispisuje tekst u radni prostor glavnog prozora

****************************************************************************/

#include <string.h>
#include "windows h"
#include "rac h"

HANDLE hInst;                              /* handle instance programa  */

HANDLE hAccTable;                          /* handle accelerator tabele */
HWND hKrajWnd;                             /* handle kraj button-a      */
HWND hwnd;                                 /* handle glavnog prozora    */


/*------------------------------------------------------------------------*/
/*  Inicijalizacija, petlja protoka poruka                                */
/*------------------------------------------------------------------------*/
int PASCAL WinMain(hInstance, hPrevInstance, lpCmdLine, nCmdShow)
HANDLE hInstance;
HANDLE hPrevInstance;
LPSTR lpCmdLine;
int nCmdShow;
{
    MSG msg;

    if (!hPrevInstance)
        if (!InitApplication(hInstance))
            return (FALSE);

    if (!InitInstance(hInstance, nCmdShow))
        return (FALSE);

    /* Glavna petlja za protok poruka */

    while (GetMessage(&msg, NULL, NULL, NULL)) {
        if (!TranslateAccelerator(hwnd, hAccTable, &msg)) {
            TranslateMessage(&msg);
            DispatchMessage(&msg);
        }
    }
    return (msg.wParam);
}
```

```
/*------------------------------------------------------------------------*/
/*  Registruje klasu prozora. Poziva se samo za prvu instancu prog        */
/*------------------------------------------------------------------------*/
BOOL InitApplication(hInstance)
HANDLE hInstance;
{
    WNDCLASS  wc;

    wc.style = NULL;
    wc.lpfnWndProc = MainWndProc;
    wc.cbClsExtra = 0;
    wc.cbWndExtra = 0;
    wc.hInstance = hInstance;
    wc.hIcon = LoadIcon(NULL, IDI_APPLICATION);
    wc.hCursor = LoadCursor(NULL, IDC_ARROW);
    wc.hbrBackground = GetStockObject(WHITE_BRUSH);
    wc.lpszMenuName =  "RacunariMenu";
    wc.lpszClassName = "RacunariClass";

    return (RegisterClass(&wc));
}

/*------------------------------------------------------------------------*/
/*  Kreira potrebne prozore-glavni i botton                               */
/*------------------------------------------------------------------------*/
BOOL InitInstance(hInstance, nCmdShow)
HANDLE hInstance;
int    nCmdShow;
{
    RECT            Rect;

    hInst = hInstance;
    hAccTable = LoadAccelerators(hInst, "RacunariAcc");

    hwnd = CreateWindow(
        "RacunariClass",
        "racunari",
        WS_OVERLAPPEDWINDOW | WS_CLIPCHILDREN,
        CW_USEDEFAULT,
        CW_USEDEFAULT,
        CW_USEDEFAULT,
        CW_USEDEFAULT,
        NULL,
        NULL,
        hInstance,
        NULL
    );

    if (!hwnd)
        return (FALSE);

    GetClientRect(hwnd, (LPRECT) &Rect);

    hKrajWnd = CreateWindow ("Button", " Kraj ",
                WS_DEFPUSHBUTTON | WS_CHILD | WS_VISIBLE,
                Rect.right/2-30,
                Rect.bottom-Rect.bottom/4,
                60,
                25,
                hwnd,
                IDC_BUTTON,
                hInst,
                NULL);

    ShowWindow(hwnd, nCmdShow);
    UpdateWindow(hwnd);
    return (TRUE);

}

/*------------------------------------------------------------------------*/
/*  Obradjuje sve poruke koje stizu ovoj aplikaciji                       */
/*------------------------------------------------------------------------*/
long FAR PASCAL MainWndProc(hwnd, message, wParam, lParam)
HWND hwnd;
unsigned message;
WORD wParam;
LONG lParam;
{
    RECT        Rect;
    HANDLE      hOldBuffer;
    FARPROC     lpProcAbout;
    PAINTSTRUCT ps;

    switch (message) {
        case WM_COMMAND:
            switch (wParam) {
                case IDM_ABOUT:
```

Listing se nastavija na sledećoj strani ►

Nastavak sa prethodne strane

```
                lpProcAbout = MakeProcInstance(About, hInst);
                DialogBox(hInst, "AboutBox", hWnd, lpProcAbout);
                FreeProcInstance(lpProcAbout);
                break;

            case IDM_EXIT:
                DestroyWindow(hWnd);
                break;

            case IDC_BUTTON:                          /* botton  kraj  */
                PostQuitMessage (0);
                break;
        }
        break;

    case WM_SETFOCUS:
        SetFocus (hKrajWnd);
        break;

    case WM_PAINT:
        BeginPaint(hWnd, (LPPAINTSTRUCT)&ps);
        RacunariPaint(hWnd, ps.hdc);
        EndPaint(hWnd, (LPPAINTSTRUCT)&ps);
        break;

    case WM_SIZE:
        InvalidateRect(hWnd, NULL, TRUE);
            MoveWindow(hKrajWnd, LOWORD(lParam)/2-30,
            HIWORD(lParam)-HIWORD(lParam)/4, 60, 25,
            TRUE);

        break;

    case WM_DESTROY:
        PostQuitMessage(0);
        break

    default:
        return (DefWindowProc(hWnd, message, wParam, lParam));
    }
  return (NULL);
}

/*------------------------------------------------------------*/
/*   Obradjuje poruke za "About" dialog box                   */
/*------------------------------------------------------------*/
BOOL FAR PASCAL About(hDlg, message, wParam, lParam)
HWND hDlg;
unsigned message;
WORD wParam;
LONG lParam;
{
    switch (message) {
        case WM_INITDIALOG:
            return (TRUE);

        case WM_COMMAND:
            if (wParam == IDOK
                || wParam == IDCANCEL) {
                EndDialog(hDlg, TRUE);
                return (TRUE);
            }
            break;
    }
    return (FALSE);
}

/*------------------------------------------------------------*/
/*   Ispisuje tekst racunari                                  */
/*------------------------------------------------------------*/
void RacunariPaint (hWnd, hDC)
HWND  hWnd;
HDC   hDC;
{
    LOGFONT    lF;
    HANDLE     hBigFont, hOldFont;
    DWORD      TE;
    WORD       TW;
    RECT       Rect;
    char       AppName[10];

     GetClientRect(hWnd, (LPRECT) &Rect);
     Rectangle(hDC,
              (short) 20,
              (short) 20,
              (short) Rect.right-20,
              (short) Rect.bottom-20);

     strcpy (AppName, "racunari");

     /* napravi veliki font */

     memset (&lF, 0, sizeof(LOGFONT));
    strcpy (lF.lfFaceName, "Modern");
    lF.lfHeight = 48;
    hBigFont = CreateFontIndirect ((LPLOGFONT) &lF);

    hOldFont = SelectObject (hDC,hBigFont);
    TE = GetTextExtent (hDC, (LPSTR) AppName, strlen(AppName));
    TW = LOWORD (TE);
    TextOut( hDC,
             (short) Rect.right/2 - TW/2,
             (short) Rect.bottom/3,
             (LPSTR) AppName,
             (short) strlen(AppName));

    SelectObject(hDC,hOldFont);
    DeleteObject(hBigFont);
```

Jezgro *Windows* programa na C-u sa svim neophodnim elementima

PROZORI I PROCESI

Tu su i neka sistemska rešenja, koja predstavljaju novine u odnosu na jednoprogramski DOS. *Dynamic Link Library* (DLL) je logika preuzeta sa „pravih" operativnih sistema i omogućuje da se formiraju biblioteke funkcija i potprograma u formi izvršnog koda. Tako potprogrami više ne moraju biti linkovani sa programom u kome se koriste. Pozivi rutina koje se nalaze u dinamičkim bibliotekama se razrešavaju u trenutku izvršavanja programa.

Program koji radi pod *Windows*-ima mora imati kreiran barem jedan prozor. Reč prozor se ovde koristi u dva konteksta: prvi označava grafički prozor, površinu na ekranu, a drugi – programski proces. Prozor je osnovna jedinica obrade u *Windows*-ima. Prozoru se upućuju sve poruke (*message*) od strane operativnog sistema. Prozor ne mora biti vidljiv na ekranu, a ipak može biti aktivan i obrađivati poruke primljene od operativnog sistema. Jedan program može imati više aktivnih prozora (procese).

Još jedna neobičnost koja se lako uočava u izvornom kodu *Windows* programa je „stil" – program liči na paskal kod. Osim ove formalne sličnosti postoji i obaveza da funkcije programa koje će biti pozivane direktno od operativnog sistema budu deklarisane kao PASCAL. Iako se za razvoj koristi standardni C kompajler, struktura programa se razlikuje od one u klasičnom DOS programiranju. U običnom C programu uvek postoji **main** funkcija kojom započinje izvršenje programa. Sličnu ulogu u *Windows* C programima ima **WinMain** funkcija. Prilikom startovanja programa kontrola se predaje **WinMain** funkciji, čiji je zadatak da izvrši inicijalizaciju i uspostavi petlju za prijem poruka.

Pri inicijalizaciji javlja se pojam instance programa. Reč je o tome da se pod *Windows*-ime jedan isti program može startovati u više primeraka. Ti primerci se onda izvršaju konkuretno u memoriji i nazivaju se instance programa. Zanimljivo je da više instanci istog programa može da deli neke resurse (menije, na primer), koji ne sadrže lokalne podatke za instancu. Inicijalizacija programa sadrži neke obavezne elemente. Potrebno je registrovati klasu prozora (**window class**) koji će se koristiti za glavni prozor programa.

TRIVIJALNI PRIMER

Primer kojim ilustrujemo pisanje *Windows* aplikacije u C u je krajnje jednostavan program koji ispisuje tekst „računari". Iako trivijalan, program sadrži sve obavezne elemente *Windows* programa i ilustruje deo koda (overhead) koji se uvek mora ispisati, čak i ako program ne radi ništa. Zbog veličine ovog obaveznog dela, za razvoj programa obično se koristi jedan ogoljeni kostur programa kao model izvornog koda. Taj model se onda kopira kao osnova svakog programa čiji se razvoj započinje. Vrlo su korisni i primeri koji se isporučuju uz SDK. Uz minimum modifikacija uklopiće se u korisnički program i uštedeti prilično vremena i truda.

Prevođenje i povezivanje programa nezamislivo je bez upotrebe programa **make**. Osim izvornog C koda, naš program sačinjavaju i neki drugi elementi, pa **make** datoteka izgleda ovako

```
all: rac.exe
rac.res: rac.rc rac.h
rc -r rac.rc
rac.obj: rac.c rac.h
cl -c -AS -Gsw -Oas -Zpe -W2 rac.c
rac.exe: rac.obj rac.def
link /NOD rac,,, libw slibcew, rac.def
rc rac.res
rac.exe: rac.res
rc rac.res
```

U izvršni kod programa uključen je i **rac**.res, koji se dobije kada se risors datoteka rac.rc prevede *Resource* kompajlerom. U njoj je opisan meni našeg programa. Tu su opisani i tasteri koji u programu dobijaju posebna značenja (**accelerators**). Tako smo, na primer, kombinaciji tastature Alt+A dodelili opciju **About** menija. Poruka koju program prima kad je pritisnuta neka od opisanih kombinacija tastera je ista kao kada je opcija izabrana iz menija. Ovde spada i opis dijaloga (**dialog box**) koji se aktivira izborom **About** opcije u meniju.

Kod poziva linkera koristi se rac.def. Sve funkcije programa koje će biti direktno pozivane od strane operativnog sistema moraju biti pobrojane u ovoj datoteci. **Winstub.exe** je program koji se linkuje sa našim programom i izvršava se ako je program startovan iz DOS-a. **Winstub** ispisuje poznatu poruku „This program requires Miscrsoft Windows". Umesto njega može se uključiti bilo koji drugi DOS program, koji će se aktivirati pod istim uslovima.

OKOSNICA PROGRAMA

Iz samog C koda izdvojićemo i opisati najvažnije delove programa. To je ujedno i uvid u strukturu programa i odnose među funkcija-



me. Osim naziva funkcije **WinMain**, koji je obavezan, ostali nazivi funkcija nisu propisani, već su stvar standarda koji pojednostavljuje prepoznavanje odgovarajućih delova programa.

WinMain: funkcija kojom počinje izvršavanje programa. Poziva funkcije za globalnu inicijalizaciju i inicijalizaciju instance (primerka) programa. U inicijalizaciji se kreira glavni prozor ovog programa i registruje procedura za obradu poruke. **WinMain** funkcija uspostavlja petlju za promet poruka (**message**) koje dostavlja operativni sistem. Poruke se u ovoj petlji ne obrađuju, već se samo omogućava i nadzira njihov protok.

InitApplication: vrši globalnu inicijalizaciju progama (samo prva instanca). U našem slučaju, samo registruje klasu prozora koja će se koristiti kao glavni prozor programa. Pri tome se registruje i procedura (**MainWndProc**) koja će obrađivati poruke te klase prozora.

InitInstance: inicijalizuje instancu (svaki primerak programa). Kreira glavni prozor i pomoćne prozore ako ih ima. Učitava globalne objekte koji će se koristiti u programu.

MainWndProc: prima poruke namenjene programu (njegovom glavnom prozoru) i poziva odgovarajuće funkcije za obradu. Poruke koje se ne obrađuju vraća sistemu na (**default**) obradu.

Navedene funkcije predstavljaju osnovu svakog *Windows* programa. To je, u stvari, minimalna količina izvornog koda koja je potrebna da program proradi. Pored toga, naš primer ima još dve funkcije:

About: obrađuje poruke namenjene „**About**" dijalogu (**dialog box**). Kako je dijalog jedna vrsta prozora, ova funkcija odgovara funkciji **MainWndProc** za glavni prozor.

RacunariPaint: iscrtava okvir i ispisuje tekst „racunari" u prostoru glavnog prozora. Pri tome koristi font dobijen uvećanjem (**CreateFontIndirect**) postojećeg sistemskog fonta.

TRUD SE ISPLATI

U ovom primeru primećujete potpuno odsustvo delova programa koji bi obrađivali rukovanje tasterima i mišem. S obzirom da smo koristili samo elementa korisničkog interfejsa koje *Windows* potpuno podržava (**meni**, **button**) za tako nešto nema ni potrebe. Do našeg programa stižu već obrađene informacije koje *Windows* prosleđuje kao poruke.

Začuđuje je da ovako jednostavan zahtev iziskuje toliko linija koda, i to može delovati pomalo obeshrabrujuće. Za divno čudo, broj linija koda mnogo sporije raste od povećanja složenosti programa. Naročito se to može reći za aplikacije koje su grafički orijentisane. Kada se uzme u obzir i korisnički interfejs i hardverska nezavisnost, trud se svakako isplati.

Tema koju smo načeli – programiranje za *Windows* – izuzetno je obimna i atraktivna, te zahteva mnogo više pažnje nego što to jedan članak može da pruži. Zato je ovaj tekst samo uvod, prva informacija o C programiranju za *Windows*. U sledećem broju upustićemo se u detaljniju analizu *Windows* programa na jednom složenijem (i upotrbljivijem) primeru.

KUĆNI IMENIK

Ovo je prvi domaći program za PC računar koji mi je došao do ruku. Unapred sam, u sebi, poželeo da to bude dobar predstavnik domaće softvaraka induatrije. Moja očekivanja su se delimično ispunila.

Program nazvan KUIM je namenjen vođenju kućnog imenika (odatle mu i ime), a može poslužiti i kao adresar i kao izvor još nekih podataka vazanih za imena iz imenika.

Paket se sastoji od jedne diskete 5.25" (bez naziva programa na nalepnici!?), uputstva za upotrebu (13 stranica A5 formate), registracione karte i koverte sa adresom izdavača. Sve je to upakovano u kartonski omot sa nazivom proizvoda i najlonski omotač.

Da me ne bi stiglo neko od Marfijevih pravila, prvo sam pročitao uputstvo. Onda sam izvršio instalaciju sa programske diskete na drugu od 3.5" jer samo one mogu u moj laptop. Pre startovanja programa pregledao sam sadržaj programske diskete. Postoji i datoteka READ ME (zašto ne PROCITAJ ME), koja sadrži neka dodatne informacije o programu.

Dakle, nakon startovanja programa, ako je u pitanju nova datoteka, pojavljuje se maska za unos podataka. Ako je u pitanju postojeća datoteka, onda je prikazana maska sa podacima iz prvog sloga u bazi podataka u režimu rada „listanje".

Sa desne strane, pored maske za unos – pregled podataka je spisak komandi kojim program raspolaže. Obzirom da je reč o klasičnoj bazi podataka, sve nophodne komanda za kontrolu programa i podataka su tu. Predviđeno je da se komande izdaju preko funkcijskoh tastera od F1 do F10. Taster F10 se i u uputstvu i u programu naziva FO (ne F-nula nego F-slovo O). To može da izazove nepotrebne nedoumice kod korisnika.

Dakle, kada je program u režimu „listanje", onda smo u mogućnosti da pregledamo bazu podataka. Tasteri koji to omogućuju su standardni Up, Down, PgUp, PgDn, Home, End. Na mojoj kopiji tasteri PgUp i PgDn nisu obrađivali posao uvek korektno. Dešava se da sa prvog sloga skoči na treći, a nikako neće na zadnji. Ako bazu listam unazad, onda preskače pretposlednji slog. Ostali tasteri funkcionišu korektno.

Ako želimo unos novih podataka, aktiviramo F1 i dobijamo praznu masku koju treba popuniti. Maske je vrlo komforna, a zgodno je što postoji mogućnost ubrzanja unosa. Naime, ako, u adresi, unesete poštanski broj grada, na primer 11000, onda će KUIM sam dodati pozivni znak za telefon – 011. Isto tako, pamti se prethodni unos, pa je moguće da se iskoristi u novoj masci i tako ubrza unošenje podataka. Možda bi trebalo da se nazivi polja u koje unosimo podatke, a koji se pojavljuju ispod maske, označe malo bolje (inverzno i slično) da bi bili uočljiviji. To bi, u početku rada, bilo od koristi. Kada unesemo nove podatke, snimićemo ih na disk sa F10.

Program omogućuje i pregled sadržaja baze podataka (F6), kao i sortiranje po abecedi (F7). Najinteresantnija je opcija za pretraživanje (F8). Čini mi se da je zgodno napravljena i



da je moguće relativno brzo naći adrese koje zadovoljavaju zadati kriterijum. One adrese koje smo izdvojili po nekom kriterijumu, možemo izdvojiti u novu datoteku, a više takvih, ili nekih drugih datoteka, možete spojiti u jednu.

Tu su, naravno, i opcije za brisanje slogova, odnosno, njihovu izmenu – korekciju.

Kao možda najkorisniju, izdvojio bih opciju štampanja. U njoj su ponuđene neke, zaista, korisne mogućnosti. Mogu se štampati memorandumi, nalepnice, spiskovi adresa i telefona i samo telefona, ličnih kartona, kao i praznih formulara za unos podataka. Izgleda da je obuhvaćeno sve što treba jednom kućnom imeniku, a i više od toga.

Najslabija tačka ovog proizvoda je uputstvo za upotrebu. Ono je toliko loše da je moguće da korisnik, naročito početnik, odbaci ceo, inače dobar, program zbog toga. U svetu se veoma mnogo polaže na izgled i sadržaj uputstava za rad sa softverom i hardverom i to ne bez razloga. Loša uputstva su čest razlog propasti nekog proizvoda na svetskom tržištu. S tog aspekta gledano, nije čudno što softverski paketi koje kupujemo na Zapadu imaju tako obimna i kvalitetna uputstva.

Obzirom na nisku cenu programa i sve veći, da ne kažem ogromni, broj računara po kućama, mislim da bi trebalo da ima dosta korisnika, odnosno, kupaca. Inače, ovaj program bi se mogao naći i u mnogim firmama koje posluju sa građanstvom ili imaju puno poslovnih partnera. Korišćenjem KUIM-a mnogi poslovi bi bili znatno ubrzani. Naravno, oni koji znaju *dBASE*, *Clipper* i slične programe će reći da mogu i sami to da naprave „za jedno popodne". Verujte mi, KUIM je toliko jeftin da se ne isplati gubiti vreme na programiranje nečega što već dobro radi.

Ja zbilja preporučujem ovaj program, pod uslovom da se uputstvo za upotrebu ozbiljno preradi.

Kontakt adresa

Jugovideo
11000 Beograd, Prote Mateje 36
Tel/fax (011) 431-848
Cena: 480 dinara

Vladimir Stamenović

KLASE – RAZGRANIČAVANJE GRANICA

Ova serija tekstova se bavi glavnim osobinama jezika C++. Veliki deo jezika C++ poklapa se sa jezikom C i te, zajedničke, osobine ovde nećemo posebno objašnjavati. C++ uvodi niz novina u odnosu na sam C i uzdiže ga na nivo objektnog jezika. Skoro svaki C program će se prevesti i pod bilo kojim C++ sistemom, ali obratno ne važi.

Duško Savić

Već u prvom primeru ovog teksta sreli smo se sa preopterećenjem operatora <<, a sada vidimo da je ista ideja primenjena i na same funkcije. Kao jedan od neophodnih uslova da je jezik objektno orijentisan, na početku smo naveli i postojanje *polimorfizma*, tj. da nam jezik dozvoljava da isto ime koristimo u različitom kontekstu. C++ ovaj uslov, dakle, ispunjava. Jedina razlika je terminološka: govorimo o „preopterećenju", a ne o polimorfizmu. Da li je to, zapravo, bitno? Naravno, ne: važno je da sve to olakšava život programeru – šta nas briga koje je boje mačka ako lovi miševa!!!

Poslednje bitno unapređenje pojma funkcije je mogućnost da se podacima preda početna vrednost direktno u deklaraciji funkcije. Recimo da je funkcija **funk** deklarisana sa

void funk(int i=0)

Možemo je zvati sa **funk()**, ili npr. sa **funk(47);**. U prvom slučaju, poziv bi bio ekvivalentan sa **funk(0)**. Ime varijable, i, je opciono u deklaraciji čak i u ovom slučaju, kada se početna vrednost postavlja direktno.

Ovakvih argumenata u listi može biti više, ali svi kojima se vrednost postavlja direktno moraju biti na kraju lista.

void funk2(int q, int, r, int v = 44, int ty=78);

Do sada smo koristili klase a da nismo videli kako se definišu. Klasa je način da se na sintaksnom nivou zajedno sa podacima spakuju i funkcije koje deluju nad tim podacima. Definisanje klase je isto što i uvođenje potpuno novog tipa podataka. U terminima sa početka ovog teksta, klasa je otelotvorenje apstraktnog tipa podataka i ne razlikuje se ni po čemu od tipa kao što je **float**. Za dva podatka tipa **float** prevodilac već zna koje operacije su moguće – e zna i kako da ih izvede. Za novi tip podatka, klasu, prevodilac „uči" od programera koje su operacije moguće i kako da ih izvede.

Sintaksno, definicija klase počinje službenom rečju **class**, iza koje dolazi ime funkcije, iza kojeg je telo funkcije (označeno vitičastim zagradama) i završava se tačkom i zapetom. Telo funkcije sastoji se iz definicija promenljivih i deklaracija funkcija. Te promenljive i funkcije su sastavni deo klase i mogu se koristiti samo u objektima koji pripadaju toj klasi.

Promenljive i funkcije unutar klase su članovi (*members*) klase. U drugim jezicima za to se upotrebljava izraz „metodi", a u C++-u je uobičajeno govoriti o pridruženim funkcijama (*member classes*). Osnovno svojstvo delova klase je lokalnost – van klase ih ne možemo koristiti. Podrazumeva se da je sve lokalno unutar klase, ali postoji i službena reč **private** kojom se lokalnost eksplicira. Ako nema službenih reči **private**, **protected** ili **public** podrazumeva se da piše **private**. Dakle sve je **private** ako se drukčije ne navede. Da bi programer ipak mogao da koristi članove klase, mora se upotrebiti službena reč **public**.

Evo jedne vrlo jednostavne klase u programu a10.cpp.

Listing 10 – a10.cpp

```
// a10.cpp :  jednostavna klasa
#include <iostream.h> // deklaracija za tok
class prvaklasa {
      int i; // podrazumeva se da je tipa private
public: // sve odavde do kraja klase je javno
      void je(int v) {i = v;} // funkcija tipa inline
      int vrati(){ return i;}  // takodje inline funkcija
};
main() {
      prvaklasa A;
      A.je(2);
      int q = A.vrati();
      cout << q << endl;
}
```

Jednostavna klasa

Definisali smo klasu **prvaklasa** tako da se sastoji iz tri člana: prvi je promenljiva **i**, a druga dva člana su funkcije **je(int v)** i **vrati()**. Promenljiva **i** je lokalna, tako da se nijedno drugo „i" nigde u programu ne bi sa tim **i** poklapalo. Vrednost lokalne promenljive **i** možemo menjati samo pomoću funkcija iste klase. Ovde su definisane dve funkcije – obe **public** – što znači da ih možemo upotrebiti i van klase u kojoj su definisane. I tu postoji ograničenje: možemo ih primeniti samo na promenljivu čiji tip se poklapa sa klasom kojoj su funkcije pridružene. U glavnom programu promenljiva **A** je tipa **jednaklasa** tako da izraz **A.je(3)** ima smisla. U sintaksnom pogledu, poziv funkcije (upućivanje poruke objektu **A**) obeležava se isto kao i da je funkcija deo struktura podataka: između imena objekta i imena funkcije postavlja se tačka. U gornjem primeru, posle naredbe

A.je(2);

vrednost interne promenljive **i** klase **jednaklasa** postala je 2, poziv

A.vrati()

očitava je u predaju nekoj drugoj promenljivoj (ovde je to **int q**).

Obe funkcije su tipa **inline**, iako se službena reč **inline** ne spominje. Takva identifikacija je moguća samo u definiciji klase – klasa je čest ali sintaksno specifičan objekat i u njoj prevodilac „može i sam da se snađe".

STANJA I OPERACIJE

Objekat ima unutrašnje stanje (vrednosti internih promenljivih) i spoljašnje operacije (pridružene funkcije). U „pravoj" objektnoj terminologiji koristili bismo i izraz „poruka" za poziv pridružene funkcije. U gornjem primeru, objektu A poslata je poruka je sa parametrom 2.

Objekat pamti svoje unutrašnje stanje i to je još jedna razlika između poziva obične i pridružene funkcije – obične uvek počinje „od nule", a pridružena deluje nad stanjem zapamćenim u internim varijablama. Ista pridružena funkcija može prilikom svakog poziva raditi nešto drugo. Pogledajmo program a11.CPP.

Listing 11 – a11.cpp

```
// a11.cpp :   izmenljiva interna stanja u objektu
#include <iostream.h> // deklaracija za tok
class drugaklasa {
      int trenutnoStanje;
public:
      void start() {trenutnoStanje = 0;}
      void next() {
            switch(trenutnoStanje) {
            case 0 : trenutnoStanje = 1; break;
            case 1 : trenutnoStanje = 2; break;
            case 2 : trenutnoStanje = 4; break;
            }
            cout << "trenutnoStanje = " << trenutnoStanje << "\n";
      }
};
main() {
      cout << "A11 CPP      izmenljive interna stanja u objektu " << endl;
      drugaklasa OK;
      OK.start();
      for (int i = 0; i < 3; i++)
            OK.next();
}
```

Izmenljiva interna stanja u objektu

Interna varijabla klase **drugaklasa** je samo jedna: **trenutnoStanje**. Vrednosti te varijable su od 0 do 4, a koja će tačno vrednost biti uneta zavisi od poziva pridružene funkcije **next()**. Početno stanje unosimo porukom **start()**. Rezultat programa je niz linija na ekranu

trenutnoStanje = 1
trenutnoStanje = 2
trenutnoStanje = 4

NAČIN KORIŠĆENJA – NAČIN RADA

Objektu upućujemo poruke da bismo postigli neki cilj. Način postizanja tog cilja i ne mora biti poznat – poruka << upućena objektu **cout** štampa nešto na ekranu, ali mi uopšte ne znamo kako se to radi. Niti nam to znanje treba – lakše je koristiti postojeće nego stalno pisati iznova! Osim u slučaju **inline** funkcija, u deklaracija klase i ne navodi kako pridružene funkcije deluju. Interfejs klase je potpuno odvojen od implementacija. U nekom budućem trenutku programer može zaželeti da promeni funkcije i to je moguće izvesti tako da interfejs ostane isti. Štaviše, biće potrebno samo da se prevede i iznova poveže samo promenjeni deo programa – a ne i ceo program (kao što je to slučaj kod drugih jezika). I u fazi pisanja programa ova razdvojenost interfejsa klase od implementacije je dobrodošla: možemo „skicirati" program i proveravati ga prevođenjem, a da same funkcije uopšte i nisu napisane. Tako postojanje klasa omogućava stvaranje prototipova programa, što je velika prednost u praksi.

KONSTRUKTORI I DESTRUKTORI

Kada definišemo primerak ugrađenog tipa podataka, npr. sa **int p**, prevodilac u memoriji odvaja prostor za novu promenljivu. Po naredbi programera, prevodilac će u taj prostor postaviti i neku vrednost. Kada tokom izvršenja programa tako definisana promenljiva „napusti" oblast svog važenja, prevodilac oslobađa rezervisanu memoriju i time uništava promenljivu. Opšta ideja u C++-u je da se novi tipovi podataka što manje razlikuju od ugrađenih, pa za svaku klasu postoje posebna funkcije za konstruisanje i brisanje pripadajućih objekata. Te funkcije su poznate pod imenima kontruktora (*constructor*) i destruktora (*destructor*), respektivno. Prilikom „oživljavanja" objekta prevodilac zove konstruktorsku funkciju, a prilikom „brisanja" – zove destruktorsku funkciju. Konstruktora može biti i više, a destruktor je uvek jedan i samo jedan. Nije obavezno da klasa sadrži ni konstruktor ni destruktor. Tada prevodilac preduzima samo minimum akcije, ali to često nije „ono pravo".

Konstruktor je pridružena funkcija i istog je imena kao i sama klasa. U skoro svim slučajevima konstruktor podrazumeva da treba zauzeti prostor za sve interne promenljive klase. Destruktor se takođe zove isto kao i klasa ali ispred njega dolazi poseban znak, tilda, ~. Destruktor nikad nema argumente, a programer ga ne može eksplicitno zvati (osim u verziji 1.0 C++-a).

Primer klase sa tri konstruktora i destruktorom dat je na listingu a12.cpp.

Listing 12 – a12.cpp

```
// a12.cpp : konstruktori i destruktor
#include <iostream.h> // deklaracije za tok
class trecaklasa {
        int i,j,k;
public:
        trecaklasa() { i = j = k = 12; } // konstruktor
        trecaklasa(int m) { i = j = k = m; } // drugi konstruktor
        trecaklasa(int a, int b, int m) {
                        i = a;
                        j = b;
                        k = m;
        } // treci konstruktor
        ~trecaklasa(){}
        void print(char * msg) {
           cout << msg << " \n";
           cout << "i = " << i << "\n";
           cout << "j = " << j << "\n";
           cout << "k = " << k << "\n";
        }
};
main() {
        trecaklasa A; // zove konstruktor bez argumenata
        trecaklasa B(56); // aktivira konstruktor sa jednim argumentom
        trecaklasa C(1, 2, 3); // konstruktor sa tri argumenta
        A.print("A -- konstruktor bez argumenata");
        B.print("B -- konstruktor sa jednim argumentom");
        C.print("C -- konstruktor sa tri argumenta");
}
```

Konstruktori i destruktor

Rezultat programa **A12.CPP** je:

A -- konstruktor bez argumenata
i = 12
j = 12
k = 12

B -- konstruktor sa jednim argumentom:
i = 56
j = 56
k = 56

C -- konstruktor sa tri argumenta:
i = 1
j = 2
k = 3

Evo primera u kojem destruktor radi nešto korisno (a13.cpp).

Listing 13 – a13.cpp

```
// a13.cpp
#include <iostream.h> // deklaracije za tok
class cetvrtaklasa {
        int i;
public:
        cetvrtaklasa() { i ++;
                         cout << "broj objekata ++: " << i << "\n";
                       }
        ~cetvrtaklasa() {
                         i --;
                         cout << "broj objekata --: " << i << "\n";
                        }
};
main() {
        cout << endl << "Program A13.CPP" << endl;
        cetvrtaklasa A;
        cout << "posle A " << endl;
        { cetvrtaklasa F; }
        cout << "posle F " << endl;
}
```

Destruktor u akciji

Rezultat ovog programa je:

broj objekata ++: 8880

posle A
broj objekata ++: 1
broj objekata --: 0

posle F
broj objekata --: 8879

Interna promenljiva i klase **cetvrtaklase** se ne postavlja ali se zatečena vrednost (**8879** za objekat **A**) uvećava za jedan, pozivom konstruktora. Za objekat **F** uvećana vrednost od **i** je **1**. Konstruktor se izvršava automatski po napuštanju oblasti važenja objekta, tako da se po izvršenju bloka u kojem se definiše sa **F** interne promenljiva **i** tog objekta smanjuje za **1**, tj. postaje **0**. Po „završetku života" objekta **A** (a to je ovde baš na samom kraju programa), ponovo se aktivira destruktor – ovoga puta za objekat A, pa se prikazuje prava početna vrednost: 8879.

STATIČKI ČLANOVI KLASE

U primeru a13.cpp postoje dva objekta, **A** i **F**, svaki sa svojom internom promenljivom **i**. Ti objekti su identični ali nezavisni. Ako objekti treba da sarađuju, tj. da pristupaju istoj internoj promenljivoj, moramo uvesti dodatnu deklaraciju službenom rečju **static**. Statički element klase zauzima memorijski prostor samo jednom, ne umnožava se kroz objekte, i obično služi za komunikaciju između objekata iste vrste. U primeru a14.cpp objekti „njuškaju" po memoriji, tj. ispituju da li imaju „živih rođaka".

Listing 14 – a14.cpp

```
// a14.cpp : staticka definicija elemenata klase
#include <iostream.h> // deklaracije za tok
class petaklasa {
        static i; // ovo automatski postavlja i na nulu
public:
        petaklasa() { i ++; }
        ~petaklasa() { i --; }
        void pronjuskaj() {
             if (i>1)
                cout << "Nisam sam!\n";
             else
                cout << "Sam sam! (Smrc, smrc...)\n";
        }
};
main() {
        cout << endl << "Program A14.CPP" << endl;
        petaklasa A;
        A.pronjuskaj();
        {
          petaklasa B;
          B.pronjuskaj();
        } // napusta oblast vazenja pa destruktor brise B
        A.pronjuskaj();
} // napusta oblast vazenja pa destruktor brise A
```

Statička definicije elemenata klase

Rezultat ovog programa je:

Sam sam! (Smrc, smrc...)

Nisam sam!
Sam sam! (Smrc, smrc...)

Prevodilac sam postavlja **0** u sve statičke elemente klase. Moguće je i ovo:

```
class bezveze
{
    static int i = 98;
}
```

a to će sintaksno i proći, ali će prevodilac rezervisati prostor za statičku promenljivu u svakoj datoteci koja sadrži deklaraciju klase. Tokom povezivanja to bi se prijavilo kao greška (osim ako se deklaracija takve klase ne pojavljuje jedan jedini put).

Spomenimo samo da u verziji 1.0 C++-a postoje i druge mogućnosti za postavljanje statičkih članova.

ZAGLAVLJE KLASE

Klasa je novi tip i podrazumeva se da njen autor želi da je koristi i u budućim programima. Deklaraciju klase treba razdvojiti od implementacije i da ne bi stalno prevodili sve delove programa: ako je tekst pridruženih funkcija u datoteci nezavisnoj od deklaracije klase, onda je za izmenu unutar klase dovoljno prevesti samo funkcije. Time se postiže zasebno prevođenje klase. Posle prevođenja, pridružene funkcije su u (skoro) prevedenom obliku na disku. Programer koji koristi takvu klasu pomoću **include** naznači zaglavlje koje klase mu je potrebno, u svom programu stvara objekte iz klase, i povezuje (*link*) sa prevedenim oblikom pridruženih funkcija negde sa diska.

U C-u je sasvim uobičajeno da se skupovi srodnih funkcija razdvajaju u posebne datoteke, sa odgovarajućim zaglavljima. U C++-u se tako *mora* raditi. Ako neka funkcija treba da ima pristup internim promenljivame klase, onda prethodno mora biti deklarisana u definiciji te klase. C takođe ne zahteva da se uvek poštuje definicija funkcije iz spoljnje biblioteke: potpuno je dozvoljeno „priručno" izmeniti definiciju funkcije u samo tekstu programa. U C++-u, to ne može tako, već se sa **include** mora eksplicitno reći odakle da se uzima zaglavlje ako se u programu koristi neka pridružena funkcija. To rešava većinu problema, ali ne i sve: ponekad je redosled navođenja datoteka u include bitan. Ovaj problem postoji u svim C++ prevodiocima po standardu do verzije 1.2.

Razdvajanje zaglavlja od teksta klase u dva datoteke povlači još jednu opasnost: da se više od jedanput uveze jedno isto zaglavlje. Ponovna deklaracija klase je greška (bar prevodioca što se tiče), pa moramo u zaglavlje ugrađivati određene provere. Tehnički, one se svode na pretprocesorske direktive **#define**, **#ifdef** i **#endif**. **#define** se obično koristi za definisanje pretprocesorskih direktiva, ali može – a to je za ovu svrhu bitnije – postavljati određene vrednosti u promenljive. Na primer:

#define NESTO2

ali postoji i oblik

#define NESTO 3.14159

U oba slučaja pretprocesor može testirati **NESTO** da vidi da li je tu postavljena neka vrednost. To se radi pomoću **#ifdef**.

#ifdef NESTO

Ako je rezultat tačan, tj. **NESTO** ima neku vrednost, naredbe iz **#ifdef** će „videti" i prevodilac i, shodno tome, prevešće ih. Kraj takve grupa naredbi je direktiva

#endif

ili

#endif NESTO

C++ ignoriše tekst iza **#endif**, tj. tekst do kraja reda je komentar.

Suprotno od **#define** je **#undef**, što će u nekoj budućoj **#ifdef** naredbi dati negativan logički rezultat. **#undef** takođe zaustavlja izvršenje makronaredbe. Konačno, postoji i **#ifndef** kao pandan direktivi **#ifdef**, i baš ta naredba je zgodna za odbranu od prekomernog uključivanja istih zaglavlja u program. Sve zajedno, šema zaglavlja je:

```
#ifndef NESTO_ZA_KLASU
#define NESTO_ZA_KLASU
// zatim definisemo klasu
#endif NESTO_ZA_KLASU
```

Umesto **NESTO_ZA_KLASU** treba stavljati posebna ali jedinstvena imena. U praksi se ispostavilo da je zgodno da ona počinju spuštenom crticom, _.

Primer klase sa zaštitom zaglavlje dat je na listingu a15.cpp.

Listing 15 – a15.cpp

```
// a15.cpp : sprecavanje ponovne definicija klasa
#ifndef _SESTAKLASA
#define _SESTAKLASA
class sestaklasa {
        int i;
public:
        sestaklasa() { i = 0; }
};
#endif _SESTAKLASA
main() {
        sestaklasa A;
}
```

Sprečavanje ponovne definicije klase

DEFINISANJE PRIDRUŽENIH FUNKCIJA

Do sada su sve funkcije bile tipa **inline**, tj. definisane „na licu mesta". U opštem slučaju, definicije funkcija su u posebnim datotekama, za svaku klasu po jedna. Da bismo definisali pridruženu funkciju, prvo moramo prevodiocu staviti do znanje da je funkcija deo klase. To se postiže simbolom za rezoluciju oblasti (**scope resolutiom operator**), koji se sastoji iz dve dvotačke, **::**. Evo jednostavnog primera (a16.cpp).

Listing 16 – a16.cpp

```
// a16.cpp : opsti nacin za definisanje pridruzenih funkcija
#include <iostream.h> // deklaracije za tok
class sedmaklasa {
        int i;
public:
        sedmaklasa(); // deklaracija funkcije
        void print(); // jos jedna deklaracija funkcije
};
// u samoj klasi se definisano nista
// prevodilac zahteva da negde kasnije nadje
// sta su funkcije sedmaklasa() i print();
// primera radi, navescemo ih odmah ali u principu
// mogle bi biti i u nekoj drugoj datoteci

sedmaklasa::sedmaklasa(){ // konstruktor
   i = 128;
}

void sedmaklasa:: print(){ // telo funkcija print
   cout << "i = " << i << endl;
}

main() {
        sedmaklasa A;
        A.print();
}
```

Opšti način za definisanje pridruženih funkcija

Operator **::** treba koristiti kadgod nije najjasnije o kojoj funkciji se radi. U primeru a17.cpp definišemo funkciju **puts**, koja ima isto ime kao i standardna **puts** funkcija iz ANSI biblioteke.

Listing 17 – a17.cpp

```
// a17.cpp : nadjacavanje imena
#include <stdio.h> // deklaracija za tok
class osmaklasa {
public:
        void puts(char near * ); // deklaracija funkcije
};
void osmaklasa :: puts(char near * tekst ){
      ::puts ("nova puts funkcija");
      ::puts (tekst);
}
main() {
        puts("Program A17 cpp");
        osmaklasa A;
        A.puts("poziv za A.puts()");
}
```

Nadjačavanje imena

Rezultat ovog programa je:

nova puts funkcija
poziv za A.puts()

Prva **puts** funkcija u glavnom programu je sasvim obična – podrazumeva se **puts** iz zaglavlja **stdio.h**. Klasa **osmaklasa** redefiniše ime **puts** i daje mu neko novo značenje. Takva funkcija može u principu raditi bilo šta, ali u ovom primeru ona se samo predstavi a zatim odštampa predati joj tekst kao i inače. Da bi se unutar te funkcije odštampao tekst, koristimo standardnu funkciju **puts**, a da bismo naznačili da se radi o globalnoj funkciji, stavili smo ispred nje simbol **::**, s tim da ime klase nismo navali.

Nastavlja se u sledećem broju

OSNOVNE KOMANDNE

Pošto smo se upoznali sa osnovama PCL-a, programskog jezika za kontrolu HP kompatibilnih laserskih štampača, i pripremili alat u vidu programa za komunikaciju računara i lasera, vreme je da upoznamo prve komanda za definisanje oblika stranice.

Dejan Ristanović

Koncepcija PCL-a se previše ne razlikuje od koncepcije kontrolnog jezika bilo kog matričnog štampača – ako pošaljete kod u intervalu [32,255], biće ispisano odgovarajuće slovo u skladu sa ASCII standardom, kod 65, na primer, proizvodi slovo A. Standardizacija u opsegu [128,255] nije ni izbliza toliko „idilična". Pre svega, za stanje modele *Hewlett Packard*-ovih štampača (originalni HP *LaserJet* i *LaserJet+*) postojala je „zabranjena zona" u opsegu [128,159] u kojoj nije mogao da se nađe ni jedan znak. Na *LaserJet* II i III ovo ograničenje je ukinuto ali se iz razloga kompatibilnosti i neizbežne „inercije" softverske industrije, razmamo retko konstruišu fontovi kod kojih je ovaj opseg popunjen. Osim toga, interpretacija svih kodova većih od 128 zavisi od izabranog seta karaktera – najčešće korišćeni PC8 sledi standard za koji su se opredelili tvorci PC računara, dok *Roman8* obezbeđuje ispisivanje većeg broja stranih slova među kojima se, na žalost, i dalje ne nalaze svi YU znaci.

KOMANDNA SEKVENCA

Kodovi iz intervala [0,31] su, najzad, označeni kao kontrolni ali je i ovaj prostor relativno slabo popunjen – definisani su samo kodovi 8 (<BS> – brisanje prethodnog znaka), 10 (<LF> – prelazak u sledeći red), 12 (<FF> – prelazak na sledeću stranu), 13 (<CR> – „laserski kurzor" na početak reda), 14, 15 (<SO> i <SI>, izbor primarnog odnosno sekundarnog seta znakova) i, naravno, 27 (<Esc>). Što se komandi <BS>, <LF>, <FF> i <CR> tiče, svaki programer koji se ikada bavio ekranom ili bilo kakvim štampačem dobro zna šta koja od njih radi. Komande <SI> i <SO> upoznaćemo kada za to dođe vreme, dok ćemo u ovim „Računarima" početi da otkrivamo zlatni rudnik sekvenci koje počinju sa <Esc>.

Opšti oblik *Escape* sekvence koju *LaserJet* prepoznaje je

<Esc> <klase komandi> <argument> <parametar>

Ako se izuzme uvodno <Esc>, čiji je kod 27 (1B heksadekadno), svi ostali delovi komande su ASCII znakovi između 32 i 128 – dok *Epson* kompatibilnom matričnom štampaču često treba slati nešto poput CHR$(27),"R",CHR$(0), laser će uvek primati komandu koju bi svaki tekst procesor lako prikazao!

U opštem opisu komandnih sekvenci PCL-a sa **<klase komandi>** smo označili niz od jednog ili (češće) 2 slova koja određuju opštu grupu komandi koje se aktiviraju – **Esc&l** (malo slovo L a ne broj 1) su komande koje se bave oblikom strane i teksta na njoj, **Esc(s** su komande za izbor fontova i tome slično. **<Argument>** je neki (potencijalno višecifreni) broj koji određuje *koliko* nečega treba uraditi dok je **<parametar>** jedno slovo koja određuje *šta* treba uraditi, odnosno koju od komandi iz izabrane **<klase komandi>** treba izvršiti. Obzirom na ozbiljne greške koje mogu nastati zbog pogrešnog tretiranja argumenta, ponovićemo da se vrednosti zadaju kao nizovi ASCII cifara – postavljanja desne margine na sedamdesetu poziciju (sekvenca **Esc&a70M**) iniciraćemo sa LPRINT CHR$(27);"&A70M", **a nikako** sa LPRINT CHR$(27);"&A";CHR$(70);"M". da bi stvar bila još interesantnija, kod nekih komandi **<argument>** može (i mora) da bude racionalan broj – kompletan rad se decimalnim tačkama (upoznali smo ih u pretprošlim „Računarima", radi se o mašinski nezavisnom zadavanju dimenzija u delovima tiče) se, na primer, zasniva na brojevima koji ne moraju da budu celi. Slanje racionalnog broja (koji može imati do četiri decimale) veoma podseća na ispisivanje po ekranu – tačka se šalje kao i svaki drugi ASCII znak, što znači da bi neka komanda mogla da glasi LPRINT CHR$(27);"&k11.75H".

Ostao nam je još **<parametar>** koji, kao što smo videli, može da bude samo slovo. Bitno je, međutim, da li je to slovo veliko ili malo – veliko slovo označava kraj komande, a malo slovo kraj segmanta u nizu komandi – komande koje pripadaju istoj **<klasi>** se, naime, mogu kombinovati čime se štedi malo štampačeve memorije i ubrzava komunikacija.

Kombinovanje komandi objasnićemo na primeru – već pomenuta sekvenca **Esc&a70M** definiše desnu marginu koja ća se nalaziti na sedamdesetoj poziciji dok **Esc&a20L** definiše levu marginu na dvadesetoj poziciji. Često nam je potrebno da istovremeno definišemo levu i desnu marginu, što bi trebalo da obavi sekvenca **Esc&a70M Esc&a20L**. Skraćivanjem ćemo uštedeti tri bajta – dovoljno je poslati **Esc&a70m20L**. Sve se, vidimo, svodi na zamenu velikog slova malim i izostavljanje polja **<Esc>** i **<klase komandi>** iz sledeće komande – ove vrednosti laser samostalno „dedukuje" pošto „zna" da se smeju kombinovati **isključivo komande iz iste klase**. Kompletna lista od dve ili više ulančanih komandi obavezno se završava velikim slovom.

Nadovezivanje komandi nismo opisali zato što smatramo da ga nema naročitog smisla konstiti – nekoliko ušteđenih bajtova prema megabajtu, dva ili tri megabajta *LaserJet*-ove memorije ne zvuči kao ušteda vredna pažnje. Ukoliko, međutim, pokušate da analizirate komunikaciju nekog komercijalnog programa sa laserskim štampačem (aktivirate, na primer, rezidentni program koji sve znakove upućene na LPT1 upisuje u datoteku i onda, pošto je rad završen, tu datoteku analizirate), nepoznavanje principa skraćivanja komandi može neograničeno da otežа i produži čitav posao.

Pošto smo upoznali opšti oblik komandi, ostaje nam samo da ih podelimo na klase i pređemo na posao. Za potrebe ove serije tekstove komanda *LaserJet*-a smo podelili na formatiranje stranice, horizontalno pozicioniranje, vertikalno pozicioniranje, ispisivanje teksta, rad sa fontovima, grafiku, makro naredbe i (neizbežne) ostale komande.

FORMATIRANJE STRANE

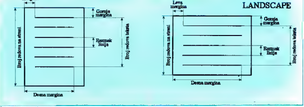
Svaki program koji kontroliše *LaserJet* kompatibilan štampač treba da započne rad slanjem niza komandi koje definišu tip papira, orijentaciju i izgled stranice. Obzirom da se dimenzije stranice najčešće zadaju u kolonama i vrstama a ne u laserskim tačkama ili inčima, na samom početku je poželjno resetovati printer i tako ga postaviti u osnovni (*Courier 10 Pitch*) neproporcionalni font. Softversko resetovanje štampača postiže se sekvencom **EscE** – ukoliko u baferu ima nešto teksta, on će biti odštampan (PCL ne obuhvata ekvivalent Epsonove sekvence Esc@, tj. ne omogućava računaru da naredi štampaču da „zaboravi" sav tekst iz memorije), poništiće se svi privremeni fontovi i makroi, biće izabran osnovni oblik stranice, osnovne margine, osnovni set karaktera i osnovni font (neke od ovih veličina mogu da se definišu pomoću kontrolnog panela).

Sledeći problem koji treba rešiti je izbor tipa papira. *LaserJet* prepoznaje umetnutu kasetu i, sasvim razložno, smatra da je u *letter* kasetu umetnut papir *letter* formata, u A4 kasetu A4 list i tome slično. U našim uslovima nije redak slučaj da se uz (u Americi ili Singapuru kupljeni) štampač dobije *letter* kaseta u koju se onda umeću A4 listovi – inč kraći radni prostor ne predstavlja poseban problem ali štampaču treba reći da njegova pretpostavka o tipu papira nije tačna. Za takvo saopštenje nadležna je komanda **Esc&l**A** pri čemu je ** označen tip papira: 1 je *executive* (7.5•10.5 inča), 2 *letter* (8.5•11 inča), 3 *legal* (8.5•14 inča), a 26 za nas najinteresantniji format A4 (210•297 mm). Vrednosti preko 80 su „zadužene" za koverte – 80 je *monarch* (3.875•7.5 inča), 81 *com-10* (4.125•9.5 inča), 90 je *international DL* (110•220 mm) a 91 *international C5* (102•229 mm). Da biste, na primer, izabrali A4 papir, treba da pošaljete komandu **Esc&l26A**.

Komanda **Esc&l**H** određuje odakle papir dolazi. *Hewlett-Packard* je, doduše, do sada proizveo samo jedan laserski štampač „za mase" koji ima više od jedne kasete, ali komanda ipak ima smisla. Vrednost 1, naime, označava da papir ulazi iz kasete, vrednost 2 znači da bez obzira na postojanje papira u kaseti, rad treba prekinuti dok senzor ne „oseti" list umetnut u uređaj za manuelno doturanje papira i tek onda štampati, a 3 označava (takođe manuelno ubačen) koverat. Definisana je i vrednost 0 koja označava nastavak rada bez promene izvora papira ali kao nuzprodukt izaziva štampanje tekuće strane (ako na njoj nečega ima) i prelazak na sledeću.

Obzirom da *HP LaserJet* i *HP LaserJet+* nisu imali ove komande a da kod *LaserJet*-a II neke od njih nisu funkcionisale baš savršeno, pomenućemo i „stari" način zadavanja dužine strane koji se i dalje često koristi zbog kompatibilnosti. Sekvencom **Esc&l**P** se, naime, zadaje

Slika 1



Osnovni elementi strane

dužina strane u redovima. Ako ste, sledeći naše savete, prethodno resetovali štampač (**EscE**), biće izabran ispis od 6 linije po inču, pa će *nn* u intervalu [1,60] označavati B5, [61,66] *letter*, [67,70] A4, a *nn* u intervalu [71,84] *legal* format (*nn=0* zahteva ponovno kalibrisanje štampača prema tipu kasete). Za standardan A4 format pogodno je poslati sekvencu **Esc&l70P**.

Sledeća faza je definisanje logičke strane, tj. rednog dela teksta na papiru. Treba odrediti margine i orijentaciju. Za razliku od prethodnih modela koji su podržavali *portrait* i *landscape* orijentaciju, *LaserJet III* podržava i *reverse portrait* odnosno *reverse landscape*. Ako list papira zamislimo kao pravougaonik stranica A i B pri čemu je A manje od B, kod *portrait* orijentacije lista tekst teče sleva udesno duž stranice A – dužina A određuje broj slova u redu a dužina B broj redova na strani; *portrait* orijentacija, sve u svemu, označava list kakav smo navikli da gledamo. U *landscape* orijentaciji dužina redova je određena dužom stranicom lista (B) dok broj redova zavisi od kreće stranice A – kao da ste okrenuli list papira za 90 stepeni i ubacili ga u pisaću mašinu sa širim valjkom. *Reverse portrait* su „naopaka" slova koja teku se desna ulevo a *reverse landscape* idu duž stranice B ali odozdo nagore. U najvećem broju realnih situacije koristi se *portrait* orijentacija, *landscape* format je zgodan kada treba imitirati neku A3 tabelu, praviti slajd ili smeštati dva A5 lista na jedan A4 a preostala dva su uglavnom kuriozitet.

Za razliku od prethodnika, *LaserJet III* omogućava kombinovanje *portrait*, *landscape*, *reverse portrait* i *reverse landscape* ispisa na istoj strani. To, međutim, ne znači da jedna strana može da ima više orijentacije – na osnovu tipa ispise koji preovlađuje na strani, morate se opredeliti za **osnovnu** orijentaciju lista papira koju ćete definisati se **Esc&l**O** pri čemu 0 označava *portrait*, 1 *landscape*, 2 *reverse portrait* a 3 *reverse landscape*: u najvećem broju slučajeva koristi se **Esc&l0O** što je i podrazumevano stanje štampača ukoliko ništa ne preduzmete. Posle toga se, komandama o kojima ćemo govoriti, tekst može ispisivati u bilo kom od ova četiri pravca i smera, ali se osnovna orijentacije strane nikada ne menja. Ukoliko biste, posle ispisa nekog teksta na strani, izvršili **Esc&l1O**, tekuće strana bi bila ispisana i štampač bi prešao na sledeću, usvajajući na njoj *landscape* orijentaciju.

Vlasnici *LaserJet*-a III su pošteđeni i brojnih muka sa fontovima. Na *LaserJet*-u II *portrait* i *landscape* fontovi su bili strogo odvojeni dok će novi model, ukoliko se pojavi zahtev za obrnutim ispisom, potpuno automatski rotirati postojeći font. Jedini problem može da izazove činjenica da je za takvo rotiranje fonta potrebno određeno vreme i (što je mnogo ozbiljnije) određena memorija. Vlasnicima neproširenog *LaserJet* III ponekad može da bude pogodno da ipak kreiraju (npr. primenom programa PORTOLAN iz paketa *Ventura Publisher* ili *Soft Craft*-ovog PORT2LAN) i pošalju *landscape* fontove jer će tako uštedeti neki bajt memorije. Takođe, ako se štampa veliki broj istih strana sa istim fontovima od koji su neki *landscape*, ima smisla definisati sve te fontova bez obzira na orijentaciju jer će se time štampanje ubrzati. Ali u najvećem broju „normalnih" situacija o orijentaciji fontova se ne mora brinuti.

Orijentacija ispisa definiše se komandom **Esc&l**P** pri čemu je ** ugao u odnosu na „normalu" izražen u stepenima. Možda će neki od budućih *LaserJet* štampača obezbediti proizvoljno zadavanje ugla – *LaserJet* III „prihvata" samo uglove 0 (*portrait*), 90 (*landscape*), 180 (*reverse portrait*) i 270 (*reverse landscape*). Na istoj strani se, kao što smo rekli, može kombinovati ispis različitih orijentacija, što znači da komanda **Esc&l**P neće** izazvati štampanje tekuće strane i prelazak na sledeću.

Za pozicioniranje logičke strane na papiru zadužene su komande **Esc&l**U** i **Esc&l**Z** – prva definiše vrh a druga levu ivicu logičke strane. Pozicija se zadaje u decimalnim tačkama, odnosno 720-tim delovima inča i zadaje se u odnosu na vrh odnosno levu ivicu papira. Zadati argument može biti protumačen i kao relativan ukoliko mu prethodi plus ili minus – tada označava **pomeranja** odgovarajuće ivice u pravcu većih vrednosti (plus) tj. nadole odnosno nadesno, ili u pravcu manjih vrednosti (minus), nagore odnosno nalevo, u oba slučaja u odnosu na trenutno važeću vrednost. Dejstvo ovih komandi je potpuno isto bez obzira na izabranu orijentaciju papira.

PODEŠAVANJE MARGINA

Za razliku od ovih ofseta, margine su (u skladu sa slikom 1) veoma zavisne od izabrane orijentacije – zbog toga je potrebno strogo slediti redosled reset-tip papira-orijentacija-margine. Margine se, kao i dužine strana, zadaju u linijama što znači da je izabrani font veoma značajan – margina od tri linije u fontu *Courier 10 Pitch* je sasvim različita od margina od tri linije u fontu *Line Printer 16.66 Pitch*. Dalji tekst pretpostavija da je štampač resetovan tj. da je izabran font *Courier 10 Pitch*.**

Gornja margina se zadaje komandom **Esc&l**E**, gde je ** broj praznih linija na vrhu strane – **Esc&l3P** definiše gornju marginu od tri linije. Vrednosti 0 i 1 su sa jedne strane nepoželjne pošto svi *LaserJet* kompatibilni laserski štampač (pa i *LaserJet* III) imaju hardversko ograničenja koje zahteva postojanje margine od bar 0.25 inča od svake ivice papira. Ukoliko, sa druge strane, pišete program koji će u potpunosti kontrolisati *LaserJet* koristeći nizove komandi idi na poziciju-štampaj, definisanja gornje margine od nula linija je veoma poželjno pošto omogućava programu da zadaje položaje u odnosu na gornju ivicu papira i definiše svoju gornju marginu – vertikalna pozicija neke tačke na stranici se uvek izražava u odnosu na definisanu gornju marginu!

Iako PCL omogućava definisanje gornje margine čak i kada na strani ima teksta (gornje margine tada praktično važi od sledeće strane pošto ispisani tekst neće biti pomeren pa će čak i kurzor ostati na ranijoj poziciji koja se može nalaziti i iznad margine), preporučujemo vam da marginu definišete na samom početku rada – tako se izbegavaju mnoge zabune.

PCL ne obuhvata komandu za definisanje donje margine – definiše se dužina strane, gornje margine i dužina teksta na strani tako da je donja margina ono što preostane (donja_margina = dužina_stranice – gornja_margina – dužina_teksta). Komandom **Esc&l**F** definišemo stranicu na kojoj se ispisuje ** linija teksta – za razumno popunjenu A4 stranicu treba poslati **Esc&l70p3e64F**.

Specificiranje gornje i donje margina ne predstavlja kraj definisanje vertikalnih mera stranice – štampač treba da zna razmak između linija ili, tačnije rečeno, visinu linije koja obuhvata kako tekst tako i prazninu. Šima može da se zadaje u broju linija po inču ili kao vertikalni indeks.

Zadavanje broja linija po inču predstavlja jednostavniji i grublji metod

OBLIK STRANE

<Esc> E	1B 45	Softverski reset štampača
<Esc> &l**A	1B 26 6C ** 41	Tip papira. 1=*Executive*, 2=*Letter*, 3=*Legal*, 26=A4, 80 81,90,91=kovert
<Esc> &l**H	1B 26 6C ** 48	Izvor papira. 0=bez promene, 1=kaseta, 2=ručno, 3=koverat
<Esc> &l**P	1B 26 6C ** 50	** linija na strani (B5 [1, 60], letter [61, 66], A4 [67, 70], legal [71, 84].
<Esc> &l0P	1B 26 6C 00 50	Postavljanje broja linija na strani prema tipu kasete
<Esc> &l**O	1B 26 6C ** 4F	Orijentacija lista 0=*portrait*, 1=*landscape*, 2=*reverse portrait*, 3=*reverse landscape*
<Esc> &l**P	1B 26 6C ** 50	Orijentacija ispisa 0=*portrait*, 90=*landscape*, 180=*reverse portrait*, 270=*reverse ndscape*.
<Esc> &l**U	1B 26 6C ** 55	Ofset leve ivice logičke strane u 1/720 inča.
<Esc> &l**Z	1B 26 6C ** 5A	Ofset vrha logičke strane u 1/720 inča.
<Esc> &l**E	1B 26 6C ** 45	Gornja margina ** linija.
<Esc> &l**F	1B 26 6C ** 46	** linije teksta na strani
<Esc> &l**D	1B 26 6C ** 44	** linija po inču Dopuštene vrednosti 1, 2, 3, 4, 6, 8, 12, 16, 24 i 48
<Esc> &l**C	1B 26 6C ** 43	Širina linije ** jedinica Jedinica je 1/48 inča
<Esc> &l**L	1B 26 6c ** 4C	Preskakanje „perforacija" 0=isključeno, 1=uključeno
<Esc> &a**L	1B 26 61 ** 4C	Leva margina ** znakova.
<Esc> &a**M	1B 26 61 ** 4D	Desna margina ** znakova u odnosu na levu ivicu papira.
<Esc> 9	1B 39	Postavljanje leve i gornje margine na nulu

Komande za oblikovanje strane

– koristi se komanda **Esc&l**D** gde ** može da ima vrednost 1, 2, 3, 4, 6, 8, 12, 16, 24 ili 48, najčešće će biti 6 ili 8. Širina linije može da se zada i kao takozvani **VMI** (*Vertical Motion Index*) tj. kao broj 48-ih delova inča – komanda je **Esc&l**C** što znači da **Esc&l6C** određuje linije širine 6/48 inča (8 linija po inču). Parametar ** ne mora da bude ceo broj – može se zadati decimalna tačka i najviše četiri decimale. Zadavanje VMI=0 može da napravi dosta problema pošto tada bilo kakvo zadavanje dimenzija u linijama (npr. gornje margine) gubi smisao takvo stanje treba maksimalno izbegavati.

Na početku ispisa ima smisla izvršiti i **Esc&l0L** čime se isključuje „automatsko preskakanje perforacije". Laserski štampač, jasno, piše na odvojenim listovima pa nema nikakvog smisla govoriti o perforaciji ali je termin zadržan po analogiji sa matričnim štampačima. Ukoliko je, naime, preskakanje perforacija uključeno, svaki pokušaj ispisa teksta na deo **ispod** definisane oblasti za tekst izaziva štampanje strane i prelazak na sledeću. Ukoliko je preskakanje perforacije onemogućeno, do automatskog štampanja strana ne dolazi nikad a tekst „ispod" donje granice se ispisuje sve dok može da stane na stranu, ako pređe fizičke margine strane jednostavno biva izgubljen. Ukoliko komandom **Esc&l**L** uključite (**=1) ili isključite preskakanje perforacije, gornja margina i dužina strane se automatski postavljaju na podrazumevane vrednosti, docnije ih treba ponovo postaviti sa **Esc&l**E** odnosno **Esc&l**F**.

Definisanje leve i desne margina je sasvim jednostavno – zadajemo ih sa **Esc&a**L** odnosno **Esc&a**M** pri čemu se u oba slučaja dimenzije zadaju u znakovima, prvom znaku u redu je dodeljen broj nula. Kako bi definisanje bilo jednoznačno, pogodno je da bude izabran neki neproporcionalni font, na primer već više puta pominjani *Courier 10 Pitch*; ukoliko je izabran proporcionalni font, *LaserJet* će definisati margine prema širini blanko (ASCII 32) karaktera. Obzirom da su u većini programa ove margine nula (docnije se komandama za pozicioniranje kurzora određuje gde će se šta štampati), uvedena je i komanda **Esc9** koja anulira eventualno postojeću gornju i levu marginu.

Iako naoko jednostavna, komanda za definisanje leve margine krije par zamki. Pokazuje se, pre svega, da stvarna pozicija prvog znaka u redu zavisi od toga kako će laser povući papir što znači da leva margina od 1 cm ne mora da znači da će prvo slovo biti udaljeno tačno centimetar od leve ivice lista; odstupanje nije veliko ali ume da namuči neiskusnog programera koji je navikao da su sve mere na laserski štampanoj stranici savršeno precizne. Zabunu može da izazove i činjenica da se pri horizontalnom pozicioniranju laserskog kurzora mere izražavaju u odnosu na levu ivicu papira a vertikalne mere u odnosu na definisanu gornju marginu – *Hewlett-Packard*-ovi programeri nisu uvek bili dosledni.

Slično vertikalnom indeksu, definisanje leve i desne margina može da bude praćeno definisanjem širine slova i to u sto dvadesetim delovima inča. Sekvencom **Esc&k**H** zahtevamo da svaki znak bude širok ** jedinica (1/120 inča) što, jasno, ima smisla samo kod neproporcionalnih fontova. Komandu **Esc&k**H**, međutim, često koristimo i kod proporcionalnih fontova jer pomoću nje zadajemo širinu blanko simbola koja je veoma značajna za uravnavanje desne ivice teksta. Iako ćemo o uravnavanju desne ivice mnogo više govoriti u jednom od sledećih nastavaka ove serije tekstova, za sada ćemo zapamtiti da broj ** može da ima najviše četiri decimale i da svako aktiviranje novog fonta vraća horizontalni inkrement na vrednost koju taj font podrazumeva.

Vidimo da *LaserJet* III ima praktično dve grupe komandi za pozicioniranje teksta na strani – najpre definisanje logičke strane a onda zadavanja margina na njoj. Jasno je da ovo dupliranje predstavlja nepotrebnu komplikaciju ali sve postaje jasnije kada se sagledaju uzroci – prethodni *Hewlett-Packard*-ovi modeli su podržavali samo komande **Esc&l**E, Esc&l**F, Esc&a**L** i **Esc&a**M**; sve one omogućavaju zadavanje dimenzija u linijama odnosno znacima što nije naročito pogodno. Zato su kod modela III uvedene nove komande u kojima figuriše decimalna tačka (možda *LaserJet IV* ponudi i zadavanje u laserskim tačkama?) ali su i stare komande morale biti zadržane zbog kompatibilnosti. Ostaje, dakle, da se opredelite za **jednu** grupu komandi i da je koristite u svojim programima, kombinacija obe može biti korisna samo kod nekih jako složenih strana koje kombinuju HP-GL i PCL grafiku.

Tabela 2 sumira sve komande koje smo upoznali ovoga meseca – spisak nije mali ali ćemo za pisanje prvih programa morati da znamo i kako se radi sa fontovima što je ujedno i tema našeg sledećeg napisa. Ovoga puta ćemo, međutim, posvetiti nešto prostora mogućim strategijama koje ćemo primenjivati pišući programe koji kontrolišu laser.

PROGRAMERSKA STRATEGIJA

Na neki način „prirodna" strategija bi zahtevala da se što je moguće više oslonimo na mikroprocesor koji je ugrađen u štampač – zašto i ta Motorola 68000 ne bi nečemu služila? To praktično znači da ćemo definisati stvarne margine, stvarni razmak između linije i slati linije teksta koje se završavaju sa <CR><LF> – komande za pozicioniranje laserskog kurzora koristimo samo kada je to zaista neophodno, na primer da bismo ostavili neku prazninu u tekstu. Dobra strana ovog rešenja je što se njegovom doslednom primenom dobija program koji svodi komunikaciju sa laserom na minimum, znajući da je komunikacija usko grlo rada sa laserom ovaj argument ima određenu težinu iako ga ne treba precenjivati – glavninu komunikacije odnosi prenos teksta i (naročito) grafika koji se ne može izbeći. Izrazito loša strana rešenja je što je debagovanje programa otežano – komunikacije računar-laser je u suštini jednosmerne što znači da nemamo nikakav način da saznamo tačnu poziciju laserskog kurzora, ako nešto nije u redu, možemo samo da gledamo listing i tražimo grešku što, kao što svaki programer zna, ume da bude veoma neprijatan i dugotrajan posao! Teškoće koje donosi debagovanje ovakvih programa su dovoljne da vam ne preporučimo primenu čitave metodologije.

Druga metodologija zahteva da pozicioniramo kurzor na željeno mesto, ispišemo slovo, pozicioniramo kurzor na sledeće mesto, ispišemo sledeće slovo i tako dalje. Dobra strana je što je debagovanje maksimalno pojednostavljeno: na svakom sumnjivom mestu možemo dodati jedno PRINT XP, YP i tako pročitati poziciju laserskog kurzora, začas uočavamo trenutak u kome je pozicija odstupila od željene i, prema tome, nalazimo grešku. Loša strana je što ispisivanje jednog bajta zahteva slanje petnaestak bajtova što je jednostavno previše – metod je nepraktičan.

Metod koji vam preporučujemo zahteva pozicioniranje kurzora na željenu poziciju i ispis kompletnog reda teksta (ako se tekst štampa u više stubaca, „red teksta" se odnosi na jedan stubac) – na ovaj način „nepotrebno" šaljemo petnaestak bajtova po redu. Znajući da strana ima jedva stotinak redova, ovo predstavlja redundancu od par kilobajta po strani što, čak i ako štampač nema proširenje memorije, predstavlja prilično zanemarljivo usporenje i beznačajan gubitak. Dobra strana metoda je što je debagovanje i dalje jednostavno: PRINT XP, YP će omogućiti da saznamo da li se red nalazi tamo gde ga očekujemo dok ćemo prostim posmatranjem primetiti da li i njemu ima dovoljno slova i da li je desna ivica korektno uravnata. Zato će ovakva metodologija biti primenjena u svim programima koje ćemo pisati sledećih meseci.

FORMIRANJE STABLA

Pisanja programa za pretraživanje direktorijuma i formiranja mape stabla zadatog diska neće nam predstavljati veliki problem, ako raščlanimo algoritam na pojedine ključne korake i vešto iskoristimo nake od postojećih funkcija, kojima DOS već raspolaže. Koristeći algoritam iz prošlog nastavka, u ovom broju pralazimo na pisanje konkretnog koda.

Jovan Skuljan

Kao što smo već rekli, stablo direktorijuma u memoriji ćemo predstaviti pomoću dve tabele. Prva (koju ćemo zvati DIRMAP) predstavlja matricu, čiji elementi opisuju *polja* raspoređena onako kako će se pojaviti i na ekranu. Zavisno od svog položaja u stablu, polje može sadržati direktorijum sa odgovarajućim vezama gore, dole, levo i desno, a može biti i prazno, ili samo sadržati prolaznu vezu gore i dole. U bilo kom slučaju, polje je predstavljeno sa *tri* bajta, čiju organizaciju smo detaljno objasnili prošlog puta. Prvi bajt nosi informaciju o tipu veze, dok preostale dva sadrže adresu imena direktorijuma, koji tom polju odgovara.

Odlučili smo se da matricu DIRMAP dimenzionišemo tako da ima *osam kolona* (a to znači osam nivoa po dubini organizacije stabla direktorijuma) i *256 redova*, što je sasvim dovoljno za gotovo sve „razumno" organizovane diskove. Ukupna memorija rezervisana za matricu iznosi 6K.

Imena svih direktorijuma ređaju se u zasebnoj listi koju ćemo zvati DIRNAMES, u *ASCIIZ formatu*. To je, u suštini, uobičajeni ASCII format, s tim što je svako ime praćeno *bajtom 00H* kao oznakom za kraj. Ukupna veličina tabele je 16K, što znači da može da primi preko hiljadu imena „pune dužine". Ovde pod „punom" dužinom podrazumevamo dvanaest bajtova (ime od osam bajtova, jedan bajt za tačku i tri za nastavak). Međutim, imena direktorijuma retko imaju nastavke, tako da im je dužina obično do osam karaktera. Takvih imena u tabelu DIRNAMES može stati čak 2000.

Uvek će tabela DIRMAP i DIRNAMES biti samo delimično popunjene. U toku samog formiranja one će se postupno popunjavati, a to znači da moramo predvideti neke promenljive, koje će stalno čuvati informaciju o tome dokle se stiglo. Na primer, tabela DIRMAP se formira red po red. Odvojićemo jednu posebnu promenljivu pod nazivom NUMROWS, koja će uvek sadržati ukupan broj iskorišćenih redova. Slično tome liste DIRNAMES se popunjava ime po ime, tako da možemo, u nekoj promenljivoj ENDNAME, pamtiti adresu iza poslednjeg imena, odnosno adresu za upis novog imena.

Sada ćemo preći na detaljniji opis i kodiranje pojedinih koraka algoritma objavljenog prošlog puta. Uz tekst objavljujemo i sve listinge sa komentarima, a na nekim stvarima ćemo se malo više zadržati. Savetujemo vam da paralelno pratite ovaj tekst, algoritam i program, ukoliko vas, naravno, interesuje *kako sve to radi*.

DA LI SE DISK ODAZIVA?

Odgovor na pitanje „Da li se disk odaziva?" može se dobiti na više načina. Mi smo se opredelili za poziv BIOS-ovog programa *Read Diskette Sectors*, koji čita sa diska zadati sektor (ili više sektora) i upisuje ih u memoriju. Nama, jasno, *nije cilj* da neki sektor prenosimo u memoriju, već samo da proverimo da li je disk prisutan. Ako je sve u redu, po povratku iz BIOS-a biće resetovan indikator prenosa (CF=0). Bilo kakve greške izazvaće setovanje indikatora prenosa (CF=1), a nama je ta informacija sasvim dovoljna za dalju akciju. Mogli bismo ići i dalje, pa analizirati sadržaj registra AH na izlazu, što bi nam omogućilo da tačno izdvojimo grešku koja se javila. Međutim, mi to nećemo ovom prilikom raditi. Ako malo razmislimo, gotovo je sasvim svejedno da li diskete u drajvu A nema uopšte, ili je oštećena. U oba slučaja, stablo direktorijuma ne možemo formirati.

Potprogram RESPOND napisali smo tako da stanje indikatora prenosa na izlazu definiše stanje ispitivanog diska. Da budemo precizniji, ispitivanju će biti podvrgnuti samo diskovi A i B, jer jedino diskete mogu biti vađene i stavljane po volji korisnika. Podrazumeva se, jasno, da potprogram RESPOND neće biti pozivan za diskove koji niti fizički ne postoje, odnosno čija oznaka prelazi preko poslednjeg drajva u sistemu. Drugim rečima, pre poziva potprograma RESPOND moramo biti sigurni da zadati disk fizički postoji u sistemu.

BIOS-ov servis *Read Diskette Sectors* poziva se preko interapta 13H (kao, uostalom, i svi ostali disk servisi u BIOS-u), pri čemu sadržaj registra AH mora biti 02H. Registar DL na ulazu nosi oznaku diska (nula za drajv A, jedan za drajv B i tako dalje), a DH označava stranu (odnosno glavu drajva). CH sadrži oznaku staze, CL oznaku sektora, AL ukupan broj sektora koji se čitaju, a ES:BX adresu bafera u memoriji gde će se sektori kopirati. Mi smo se opredelili da čitamo sektor *1* u stazi *0* na strani *0*.

OSNOVNI DIREKTORIJUM

Prvi korak u formiranju stabla direktorijuma jeste zapis *osnovnog direktorijuma* u tabelama DIRMAP i DIRNAMES. Posao oko toga je prilično jednostavan i obavlja ga potprogram FORMROOT. U prva tri bajta liste DIRNAMES upisuje se ime diska praćeno sa dve tačke i nulom kao oznakom za kraj imena. Prva sledeća adresa u listi biće kasnije iskorišćena za upis novog imena, a promenljiva ENDNAME služiće kao pokazivač.

U matrici DIRMAP, osnovni direktorijum zauzima polje u prvom redu i u prvoj koloni. Polje samo treba da sadrži ime (setovan najviši bit u bajtu koji nosi informaciju o tipu veze). Adresa imena je upravo početak liste DIRNAMES. Sva ostala polja matrice DIRMAP treba isprazniti, a promenljivoj NUMROWS dodeliti vrednost 0001H, jer je, nakon upisa osnovnog direktorijuma, iskorišćen samo jedan red matrice.

DA LI NIVO POSTOJI?

Program za formiranje stabla direktorijuma radi u ciklusu po nivoima matrice DIRMAP. Nakon formiranja osnovnog direktorijuma (prva kolona, odnosno nulti nivo), biće potraženi poddirektorijumi koji su u njemu sadržani. Tako će biti formirana druga kolona matrice (nivo 1). U sledećem koraku, biće ispitan svaki od tih poddirektorijuma, sa ciljem da se utvrdi dalje grananje, a to znači postojanje sledeće kolone (sledećeg nivoa). Izlaz iz ciklusa, odnosno izlaz iz programa FORMTREE uslediće kada ni jedan poddirektorijum iz tekućeg nivoa ne bude pokazivao grananje.

Kao jednostavan indikator postojanja novog nivoa služiće nam promenljiva NEWLEVEL. Na početku svakog koraka njena vrednost postavlja se na nulu, što znači da nema daljeg grananja. Ako bilo koji od direktorijuma na tekućem nivou bude imao poddirektorijume, promenljiva NEWLEVEL dobiće vrednost 01H, što znači da ciklus treba ponoviti i za sledeći nivo.

LISTA PODDIREKTORIJUMA

Na koji način ćemo proveriti da li neki direktorijum sadrži u sebi poddirektorijume? Razume se, i tu ćemo koristiti potencijale samog DOS-a. Na raspolaganju nam stoje dve moćne funkcije za pretraživanje: *Start File Search* i *Continue File Search*, koje pozivamo uobičajenim putem preko interapta 21H, koristeći za prvu funkciju kod 4EH, a za drugu 4FH.

Radi se, zapravo, o dve operacije koje mogu naći primenu kada god pretražujemo neki direktorijum, bilo da tražimo poddirektorijume ili datoteke. Da bi cela stvar funkcionisala, moramo prethodno negde u memoriji formirati ASCIIZ zapis kompletne staze koja vodi do zadatog direktorijuma, poddirektorijuma, ili datoteteke. Moguća je upotreba uobičajenih džokera (zvezdica i znak pitanja), pa tako na primer, ako zadamo stazu *A:\TEKSTOVI*.R72*, tražićemo sve datoteke koje imaju nastavak *R72*.

Adresu staze treba upisati u DS:DX, a atribut za pretraživanje u CX. Poddirektorijumi imaju atribut 10H. Pozivamo zatim funkciju 4EH, da bismo pronašli prvi element koji odgovara zadatom opisu. Svaki sledeći ćemo dobiti pozivom funkcije 4FH, bez bilo kakvih ulaznih parametara.

Dva pitanja ovde moramo objasniti. Prvo, šta znači to da *dobijamo neki element* i drugo, šta će biti ako nema *elemenata* koji bi odgovarali zadatom opisu?

Listing 1

Potprogram FORMTREE formira mapu stabla direktorijuma u tabeli DIRMAP i spisak imena poddirektorijuma u tabeli DIRNAMES. Na ulazu, ASCII oznaka diska se nalazi u promenljivoj DISK. Tabela DIRMAP podeljena je na osam kolona, koje odgovaraju nivoima poddirektorijuma, i 256 redova. Svako polje je veličine tri bajta. Prvi bajt sadrži kodiranu informaciju o tipu veze datog poddirektorijuma sa ostalima u stablu, dok druga dva bajta sadrže adresu imena poddirektorijuma u tabeli DIRNAMES. Tip veze kodira se tako da bit 7 označava da li je polje prazno ili sadrži direktorijum, a bitovi 6, 2, 1 i 0 daju redom veze udesno, ulevo, nagore i nadole. Bitovi 5, 4 i 3 se ne koriste i uvek su resetovani. Ako je stablo uspešno formirano, sadržaj promenljive TREEFLAG na izlazu je 0001H. Ali, ako je stablo previše složeno (ima više od osam nivoa uključujući i osnovni direktorijum, ili ima više od 256 redova, ili sva imena ne staju u tabelu DIRNAMES), TREEFLAG na izlazu sadrži vrednost 0002H a tabele DIRMAP i DIRNAMES ostaju nedovršene. Ukoliko se disk ne odaziva, TREEFLAG na izlazu dobija vrednost 0000H.

```
FORMTREE  PROC NEAR
          MOV  TREEFLAG,0001H     ;Pretpostavka: Stablo će uspeti.
          CALL RESPOND            ;Da li se disk odaziva?
          JNC  ROOTDIR            ;Ako se odaziva, idi napred.
          DEC  TREEFLAG           ;Signal: Disk se ne odaziva.
          RET                     ;Povratak.
ROOTDIR:  CALL FORMROOT           ;Formiraj zapis osnovnog direktorijuma.
          MOV  NEWLEVEL,01H       ;Signal: Nulti nivo postoji.
          MOV  LEVEL,0FFFFH       ;Početna vrednost brojača nivoa je -1.
NEXTLVL:  INC  LEVEL              ;Pređi na sledeći nivo.
          CMP  NEWLEVEL,00H       ;Da li nivo uopšte postoji?
          JNE  TOPROW             ;Ako postoji, idi napred.
          RET                     ;Povratak.
TOPROW:   MOV  ROWDIR,0FFFFH      ;Početna vrednost brojača redova je -1.
          MOV  NEWLEVEL,00H       ;Signal: Nije dostignut novi nivo.
NEXTROW:  INC  ROWDIR             ;Pređi na sledeći red.
          CMP  ROWDIR,00FFH       ;Da li je taj red unutar tabele?
          JA   NEXTLVL            ;Ako nije, idi na sledeći nivo.
          CALL GETDIR             ;Uzmi direktorijum sa tekućeg polja tabele.
          MOV  AL,[SI]            ;Uzmi informaciju o tipu veze.
          ROL  AL,1               ;Da li polje sadrži direktorijum?
          JNC  NEXTROW            ;Ako ne sadrži, idi na sledeći red.
          CALL GETPATH            ;Formiraj stazu za pristup direktorijumu.
          CALL PUTASTER           ;Dodaj tekst "*.*" i marker 00H.
          CALL GETLIST            ;Formiraj listu poddirektorijuma.
          JC   COMPLEX            ;Skok ako tabela DIRNAMES nije dovoljna.
          MOV  CX,DIRNUM          ;Uzmi broj poddirektorijuma.
          MOV  CX,CX              ;Da li ima poddirektorijuma?
          JE   NEXTROW            ;Ako nema, pređi na sledeći red.
          CMP  LEVEL,0007H        ;Da li ima mesta u tabeli za novi nivo?
          JNB  COMPLEX            ;Ako nema, idi napred.
          CALL CONNECT            ;Dodaj jedan poddirektorijum iznesna.
          MOV  NEWLEVEL,01H       ;Signal: Dostignut je novi nivo.
          DEC  CX                 ;Da li ima još poddirektorijuma?
          JE   NEXTROW            ;Ako nema, pređi na sledeći red.
          ADD  CX,NUMROWS         ;Nova vrednost za NUMROWS.
          MOV  CX,CX              ;Da li je nova vrednost preko 00FFH?
          JNS  COMPLEX            ;Ako jeste, idi napred.
          CALL INSROWS            ;Umetni nove redove u matricu.
          CALL MAKEDIRS           ;Upiši i poveži nove poddirektorijume.
          JMP  NEXTROW            ;Pređi na sledeći red.
COMPLEX:  INC  TREEFLAG           ;Signal: Stablo je previše složeno.
          RET                     ;Povratak.
FORMTREE  ENDP
```

Potprogram RESPOND proverava da li se zadati disk odaziva na čitanje. Ustvari, provera se vrši samo ukoliko se radi o diskovima A i B, dok se za sve druge oznake povratak vrši neposredno. Na izlazu, indikator prenosa je resetovan (NC) ako je sve bilo u redu (disk se odazivа), a setovan (CY) ako se disk ne odaziva.

```
RESPOND   PROC NEAR
          MOV  DL,DISK            ;Uzmi ASCII oznaku diska.
          CMP  DL,'B'             ;Da li je to neki od diskova A ili B?
          JBE  DISKTRY            ;Ako jeste, idi napred.
          RET                     ;Povratak (NC). Oznaka diska je iznad B.
DISKTRY:  SUB  DL,41H             ;Neka oznake počinju od nule.
          XOR  DH,DH              ;Izaberi gornju stranu diskete.
          MOV  CX,0001H           ;Staza 00H, sektor 01H.
          LEA  BX,SECTOR          ;Adresa bafera za upis sektora.
          MOV  AX,0201H           ;Priprema za poziv BIOS-a.
          INT  13H                ;Pročitaj jedan sektor sa diskete.
          RET                     ;Povratak.
RESPOND   ENDP
```

Potprogram FORMROOT formira zapis osnovnog direktorijuma u matrici DIRMAP i u tabeli DIRNAMES. Na ulazu, ASCII oznaka diska se nalazi u promenljivoj DISK. Na izlazu, potpuni naziv osnovnog direktorijuma (ime diska i dve tačke) nalazi se na adresi DIRNAMES u ASCIIZ formatu, a ENDNAME daje adresu za upis sledećeg direktorijuma (iza markera 00H u listi DIRNAMES). Formiran je odgovarajući zapis direktorijuma u prva tri bajta tabele DIRMAP, a ostala polja su ispražnjena (00H na mestu oznake za tip veze). Promenljiva NUMROWS sadrži vrednost 0001H što predstavlja ukupan broj korišćenih redova matrice DIRMAP.

```
FORMROOT  PROC NEAR
          LEA  BX,DIRNAMES        ;Uzmi adresu liste DIRNAMES.
          MOV  AL,DISK            ;Uzmi oznaku diska.
          MOV  AH,3AH             ;Iza imena idu dve tačke.
          MOV  DI,BX              ;Adresa na koju će se upisati ime.
          STOSW                   ;Formiraj zapis osnovnog direktorijuma.
          XOR  AL,AL              ;Oznaka kraja 00H.
          STOSB                   ;Označi kraj imena direktorijuma.
          MOV  ENDNAME,DI         ;Adresa za sledeći direktorijum.
          LEA  DI,DIRMAP          ;Uzmi adresu tabele DIRMAP.
          MOV  DL,10000000B       ;Pripremi kod osnovnog direktorijuma.
          MOV  [DI],DL            ;Upiši kod osnovnog direktorijuma u tabelu.
          MOV  [DI+1],BX          ;Upiši adresu imena u tabelu DIRMAP.
          MOV  DI,0003H           ;Adresa polja do osnovnog direktorijuma.
          MOV  CX,6141            ;Sva preostala polja u tabeli biće prazna.
          REPZ STOSB              ;Upiši nule u sva preostala polja.
          MOV  NUMROWS,0001H      ;Ukupan broj korišćenih redova.
          RET                     ;Povratak.
FORMROOT  ENDP
```

Potprogram GETDIR nalazi adresu tekućeg polja tabele DIRMAP. Na ulazu, tekući nivo je u varijabli LEVEL a tekući red u varijabli ROWDIR. Na izlazu, indeksni registar SI sadrži traženu adresu.

```
GETDIR    PROC NEAR
          MOV  AX,ROWDIR          ;Uzmi tekući broj reda.
          MOV  DX,LEVEL           ;Uzmi tekući nivo.
          CALL DIRADDR            ;Formiraj u SI adresu tekućeg polja.
          RET                     ;Povratak.
GETDIR    ENDP
```

Potprogram DIRADDR nalazi adresu tekućeg polja tabele DIRMAP. Na ulazu, tekući nivo je u registru DX, a tekući red u registru AX. Na izlazu, indeksni registar SI sadrži traženu adresu.

```
DIRADDR   PROC NEAR
          CALL ADDRROW            ;Formiraj u SI adresu početka tekućeg reda.
          ADD  SI,DX              ;Konačna adresa zadatog polja dobija se
          ADD  SI,DX              ;kada se na početnu adresu reda doda
          ADD  SI,DX              ;utrostručena vrednost nivoa.
          RET                     ;Povratak.
DIRADDR   ENDP
```

Potprogram ADDRROW računa adresu početka zadatog reda matrice DIRMAP. Na ulazu broj nivoa je zadat u registru AX (AH=0). Adresa na izlazu smeštena je u SI.

```
ADDRROW   PROC NEAR
          LEA  SI,DIRMAP          ;Uzmi početnu adresu tabele DIRMAP.
          MOV  CL,24              ;Svaki red ima 24 bajta (8 nivoa po 3).
          MUL  CL                 ;Formiraj relativnu adresu tekućeg reda.
          MOV  SI,AX              ;Formiraj apsolutnu adresu tekućeg reda.
          RET                     ;Povratak.
ADDRROW   ENDP
```

Potprogram GETPATH formira na adresi DIRPATH stazu za pristup tekućem direktorijumu. Na ulazu i izlazu, registar SI sadrži adresu odgovarajućeg polja u tabeli DIRMAP. Formirana staza se završava obrnutom kosom crtom, a promenljiva ENDPATH na izlazu sadrži adresu iza obrnute kose crte na kraju staze.

```
GETPATH   PROC NEAR
          MOV  LIMIT,0FFFFH       ;Nema ograničenja pri kopiranju imena.
          PUSH SI                 ;Sačuvaj adresu tekućeg direktorijuma.
          MOV  DL,5CH             ;Pripremi obrnutu kosu crtu za kasnije.
          LEA  DI,DIRPATH         ;Uzmi adresu bloka za upisivanje staze.
          XOR  CX,CX              ;Početna vrednost brojača nivoa.
BACKTREE: LODSB                   ;Uzmi informaciju o tekućem tipu veze.
          MOV  BX,[SI]            ;Uzmi adresu imena iz polja matrice.
          PUSH BX                 ;Ubaci adresu na vrh mašinskog steka.
          INC  CX                 ;Uvećaj brojač nivoa.
          AND  AL,10111111B       ;Poništi vezu udesno ako postoji.
          CMP  AL,10000000B       ;Da li je to bio osnovni direktorijum?
          JE   MAKEPATH           ;Ako jeste, idi napred na formiranje staze.
FINDTOP:  MOV  AL,00000100B       ;Da li postoji veza ulevo?
          JNZ  GOLEFT             ;Ako postoji, izađi iz ciklusa.
          SUB  SI,0019H           ;Pomeri se na polje iznad tekućeg.
          LODSB                   ;Uzmi informaciju o tipu veze.
          JMP  FINDTOP            ;Vrati se nazad u ciklus.
GOLEFT:   SUB  SI,0004H           ;Pomeri se na polje levo od tekućeg.
          JMP  BACKTREE           ;Vrati se nazad u ciklus.
MAKEPATH: POP  BX                 ;Skini sa steka adresu imena.
          CALL MOVETEXT           ;Upiši ime direktorijuma u stazu.
          MOV  [DI-1],DL          ;Dodaj obrnutu kosu crtu iza imena.
          LOOP MAKEPATH           ;Ponovi ciklus za sve poddirektorijume.
          MOV  ENDPATH,DI         ;Adresa iza poslednje obrnute kose crte.
          POP  SI                 ;Obnovi adresu tekućeg direktorijuma.
          RET                     ;Povratak.
GETPATH   ENDP
```

Potprogram MOVETEXT prenosi proizvoljan tekst u ASCIIZ formatu sa adrese BX na adresu DI, u smeru rastućih adresa. Sadržaj promenljive LIMIT na ulazu daje gornju granicu adresa koju registar DI u toku kopiranja ne sme dostići. Na izlazu, ako je sve proteklo bez problema, indeksni registar DI pokazuje iza markera 00H na kraju novoformirane kopije teksta, a indikator prenosa je resetovan (NC). Međutim, ako je gornja granica adresa dostignuta, kopiranje se prekida, a indikator prenosa na izlazu je setovan (CY).

```
MOVETEXT  PROC NEAR
          CMP  DI,LIMIT           ;Da li je dostignuta gornja granica?
          JB   MOVECHAR           ;Ako nije, idi napred.
          STC                     ;Signal: Dostignuta je gornja granica.
          RET                     ;Povratak (CY).
MOVECHAR: MOV  AL,[BX]            ;Uzmi karakter iz teksta.
          INC  BX                 ;Pomeri se na sledeći karakter.
          STOSB                   ;Upiši karakter na odredište.
          AND  AL,AL              ;Da li je to bila oznaka kraja 00H?
          JNZ  MOVETEXT           ;Ako nije, ponovi ciklus.
```

Listing se nastavlja na sledećoj strani ▶

Nastavak sa prethodne strane

```
            RET                         Povratak (NC).
MOVETEXT    ENDP
```

Potprogram PUTASTER kopira tekst "*.*" iza poslednjeg poddirektorijuma u bloku DIRPATH čime se priprema staza za pristup disku. Adresa na koju se kopiranje vrši data je na ulazu u promenljivoj ENDPATH. Na izlazu, cela staza je praćena bajtom 00H (ASCIIZ). Registar SI ostaje očuvan.

```
PUTASTER    PROC NEAR
            PUSH SI                     Sačuvaj adresu tekućeg direktorijuma.
            LEA  SI,ASTER               Uzmi adresu teksta "*.*"
            MOV  DI,ENDPATH             Uzmi adresu odredišta.
            MOV  CX,0004H               Dužina teksta iznosi četiri bajta.
            REPZ MOVSB                  Kopiraj tekst zajedno sa markerom 00H.
            POP  SI                     Obnovi adresu tekućeg direktorijuma.
            RET                         Povratak.
PUTASTER    ENDP
```

Potprogram GETLIST formira listu poddirektorijuma za dati direktorijum. Na ulazu, kompletna staza za pretraživanje zadatog direktorijuma (ASCIIZ) je na adresi DIRPATH. Ukoliko ima poddirektorijuma, oni će biti dopisani u listu DIRNAMES zajedno sa markerima 00H, a početna adresa prvog poddirektorijuma biće smeštena u DIRPTR. Ukupan broj poddirektorijuma nalaziće se u promenljivoj DIRNUM a ENDNAME će pokazivati iza poslednjeg markera 00H. Ukoliko poddirektorijuma nema, DIRNUM će sadržati nulu, a lista DIRNAMES i promenljiva ENDNAME ostaće nepromenjene. U oba slučaja indikator prenosa će biti resetovan (NC). Međutim, ako direktorijume ima, a u listi DIRNAMES nema dovoljno mesta za njihov upis, na izlazu će biti setovan indikator prenosa (CY).

```
GETLIST     PROC NEAR
            LEA  AX,DIRNAMES            Uzmi adresu liste imena direktorijuma.
            ADD  AX,4000H               Formiraj adresu iza liste.
            MOV  LIMIT,AX               To je gornja granica za upis imena.
            MOV  DIRNUM,0000H           Pretpostavka: Nema poddirektorijuma.
            MOV  DI,ENDNAME             Adresa za nastavak liste imena.
            MOV  DIRPTR,DI              To će biti početak liste poddirektorijuma.
            LEA  DX,DIRPATH             Uzmi adresu staze.
            MOV  CX,0010H               Pripremi atribut za poddirektorijume.
            MOV  AH,4EH                 Priprema za poziv DOS-a.
            INT  21H                    Pronađi prvu datoteku.
            CMP  AX,0012H               Da li je datoteka nađena?
            JNE  GETEACH                Ako jeste, idi napred
            RET                         Povratak (NC). Nema poddirektorijuma.
GETEACH:    MOV  SI,009EH               Adresa imena datoteke u bloku DTA.
            MOV  AL,[SI-9]              Uzmi atribut datoteke
            AND  AL,10H                 Da li je to poddirektorijum?
            JZ   NEXTFILE               Ako nije, idi napred
            MOV  AL,[SI]                Uzmi prvi karakter imena.
            CMP  AL,2EH                 Da li je to tačka?
            JE   NEXTFILE               Ako jeste, idi napred.
            INC  DIRNUM                 Uvećaj brojač poddirektorijuma.
            CALL MOVETEXT               Prenesi ime u listu
            JNC  NEXTFILE               Skok ako je ime upisano u listu.
            RET                         Povratak (CY). Nema mesta za upis imena.
NEXTFILE:   MOV  AH,4FH                 Priprema za poziv DOS-a.
            INT  21H                    Pronađi sledeću datoteku
            CMP  AX,0012H               Da li je datoteka nađena?
            JNE  GETEACH                Ako jeste, idi nazad u ciklus.
            MOV  ENDNAME,DI             Formiraj vrednost pokazivača.
            RET                         Povratak (NC)
GETLIST     ENDP
```

Potprogram CONNECT dodaje tekućem direktorijumu u matrici DIRMAP jedan poddirektorijum u prvom polju desno. Na ulazu i izlazu, registar SI pokazuje tekući direktorijum u matrici DIRMAP. Unapred je već provereno da ima mesta u matrici za novi nivo. Promenljiva DIRPTR na ulazu daje adresu imena poddirektorijuma.

```
CONNECT     PROC NEAR
            MOV  AL,01000000B           Maska za vezu udesno.
            OR   [SI],AL                Dodaj tekućem direktorijumu vezu udesno.
            MOV  AL,10000100B           Poddirektorijum sa vezom ulevo.
            MOV  [SI+3],AL              Upiši poddirektorijum u matricu.
            MOV  DX,DIRPTR              Uzmi adresu imena poddirektorijuma i
            MOV  [SI+4],DX              upiši je u odgovarajuće polje matricu.
            RET                         Povratak
CONNECT     ENDP
```

Potprogram INSROWS umeće nove redove u matricu DIRMAP, za upis novih poddirektorijuma. Broj umetnutih redova je za jedan manji od broja poddirektorijuma, koji na ulazu daje promenljiva DIRNUM. Redovi se umeću iza tekućeg (ROWDIR) tako što se preostali deo tabele pomera naniže. Novoformirana polja se popunjavaju "praznim direktorijumima" uz očuvanje vertikalnih veza. Na ulazu ukupan broj korišćenih redova matrice DIRMAP nalazi se u promenljivoj NUMROWS. Unapred je već provereno da u matrici ima mesta za zadati broj novih redova. Na izlazu je očuvan sadržaj SI (adresa tekućeg direktorijuma) a sadržaj promenljive NUMROWS je uvećan za broj umetnutih redova. Ako se novi redovi dodaju iza poslednjeg korišćenog reda, jedini posao potprograma je formiranje novog sadržaja NUMROWS.

```
INSROWS     PROC NEAR
            PUSH SI                     Sačuvaj adresu tekućeg direktorijuma.
            CALL MOVEROWS               Pomeri blok iza tekućeg reda naniže.
            JC   EXITINS                Skok ako se novi redovi samo dodaju.
            MOV  CX,AX                  Uzmi broj umetnutih redova u CX, a zatim
            ADD  CX,CX                  formiraj ukupan broj umetnutih polja,
            ADD  CX,CX                  množenjem broja redova sa osam (svaki
            ADD  CX,CX                  red sadrži osam nivoa)
EACHCELL:   XOR  AL,AL                  Oznaka za prazno polje
            MOV  [SI],AL                Upiši prazno polje u matricu.
            MOV  AL,[SI-18H]            Uzmi tip veze iz polja odmah iznad.
            ROR  AL,1                   Da li je tamo prisutna veza nadole?
            JNC  NEXTCELL               Ako nije, idi napred.
            MOV  AL,00000011B           Maska za vezu nagore i nadole.
            OR   [SI],AL                Formiraj vertikalnu vezu
NEXTCELL:   ADD  SI,0003H               Pomeri se na sledeće polje
            LOOP EACHCELL               Ponovi ciklus za sva umetnuta polja.
EXITINS:    POP  SI                     Obnovi adresu tekućeg direktorijuma.
            RET                         Povratak
INSROWS     ENDP
```

Potprogram MOVEROWS obavlja pomeranje naniže bloka redova iza tekućeg, kako bi se omogućilo umetanje novih. Ulaz je isti kao i u INSROWS. Na izlazu, registar SI sadrži adresu prvog bajta oslobođenog prostora (ispod tekućeg reda), a BX daje broj oslobođenih (umetnutih) redova. Indikator prenosa je na izlazu resetovan (NC) ako su novi redovi zaista umetnuti. Međutim, ako je tekući red ujedno i poslednji korišćeni red tabele, novi redovi se samo dopisuju. U tom slučaju je indikator prenosa na izlazu setovan (CY), a sadržaj SI i BX nisu od značaja. U oba slučaja sadržaj promenljive NUMROWS je uvećan za broj umetnutih (dodatih) redova.

```
MOVEROWS    PROC NEAR
            MOV  AX,ROWDIR              Uzmi tekući broj reda.
            INC  AX                     Pomeri se na red ispod tekućeg,
            MOV  DX,AX                  upamti njegov redni broj za kasnije i
            CALL ADDRROW                formiraj adresu njegovog početka u SI
            MOV  AX,NUMROWS             Uzmi broj korišćenih redova.
            MOV  BX,DIRNUM              Uzmi ukupan broj poddirektorijuma i
            DEC  BX                     formiraj broj redova za umetanje.
            ADD  NUMROWS,BX             Dodaj umetnute redove ukupnom broju.
            CMP  DX,AX                  Da li je tekući red poslednji u tabeli?
            JB   CONTMOVE               Ako nije, idi napred
            STC                         Signal: Redovi se ne umeću, već se dodaju.
            RET                         Povratak (CY)
CONTMOVE:   MOV  DX,SI                  Prenesi adresu toktakućeg reda u DX.
            CALL ADDRROW                Formiraj u SI adresu iza poslednjeg reda.
            MOV  CX,SI                  Prenesi tu adresu u CX i formiraj
            SUB  CX,DX                  ukupnu dužinu bloka koji se pomera.
            DEC  SI                     Postavi SI na poslednji bajt bloka.
            MOV  AX,BX                  Uzmi broj redova za umetanje
            MOV  DL,18H                 Svaki red ima 24 bajta.
            MUL  DL                     Ukupan broj bajtova koji se umeću.
            MOV  DI,SI                  Prenesi u DI poslednju adresu bloka i
            ADD  DI,AX                  formiraj novu vrednost poslednje adrese
            STD                         Kopiranje u smeru opadajućih adresa.
            REPZ MOVSB                  Pomeri blok na novo mesto.
            CLD                         Obnovi smer rastućih adresa.
            INC  SI                     Pomeri se na prvi bajt umetnutog bloka.
            CLC                         Signal: Redovi su umetnuti u tabelu.
            RET                         Povratak (NC)
MOVEROWS    ENDP
```

Potprogram MAKEDIRS upisuje i povezuje nove poddirektorijume u umetnute redove matrice DIRMAP. Na ulazu, indeksni registar SI daje adresu u tabeli DIRMAP direktorijuma čije grananje se trenutno vrši, a DIRPTR i DIRNUM sadrže adresu i broj poddirektorijuma u listi DIRNAMES.

```
MAKEDIRS    PROC NEAR
            MOV  AL,00000001B           Maska za vezu nadole.
            OR   [SI+3],AL              Dodaj prvom poddirektorijumu vezu nadole
            ADD  SI,0018H               Pomeri se na drugi poddirektorijum
            MOV  CX,DIRNUM              Uzmi ukupan broj poddirektorijuma i
            DEC  CX                     uračunaj da je prvi već formiran
            MOV  BX,DIRPTR              Uzmi adresu liste sa imenima.
            MOV  AH,10000011B           Poddirektorijumi imaju veze gore i dole.
EACHDIR:    CALL SKIPNAME               Formiraj u BX adresu sledećeg imena.
            MOV  [SI],AH                Upiši tip veze u prvi bajt polja.
            MOV  [SI+1],BX              Upiši adresu imena u druga dva bajta.
            ADD  SI,0018H               Pomeri se na polje ispod.
            LOOP EACHDIR                Ponovi ciklus za sve poddirektorijume
            MOV  AL,11111110B           Maska za brisanje veze nadole.
            AND  [SI-18H],AL            Ukloni vezu nadole iz poslednjeg polja.
            RET                         Povratak.
MAKEDIRS    ENDP
```

Potprogram SKIPNAME postavlja sadržaj BX na adresu sledećeg imena u ASCIIZ listi

```
SKIPNAME    PROC NEAR
            MOV  AL,[BX]                Uzmi karakter iz imena.
            INC  BX                     Pomeri se na sledeći karakter
            AND  AL,AL                  Da li je to bila oznaka za kraj (00H)?
            JNZ  SKIPNAME               Ako nije, vrati se nazad u ciklus.
            RET                         Povratak.
SKIPNAME    ENDP
```

Rutine za pretraživanje direktorijuma i formiranje mape stabla zadatog diska

Listing 2

```
Konstante, promenljive i tabele.

SECTOR    DB   1024 DUP(?)    ;Bafer za upis sektora sa diska.
DIRMAP    DB   6144 DUP(?)    ;Matrica stabla direktorijuma.
DIRNAMES  DW   10304 DUP(?)   ;Lista imena direktorijuma.
DIRPATH   DW   128 DUP(?)     ;Staza za pristup disku.
ASTER     DW   '*.*',00H      ;Maska za pretraživanje direktorijuma.
ENDNAME   DW   ?              ;Adresa iza poslednjeg imena u listi.
LIMIT     DW   ?              ;Granična adresa za kopiranje imena.
FIRPTR    DW   ?              ;Adresa imena prvog poddirektorijuma.
DIRNUM    DW   ?              ;Ukupan broj poddirektorijuma.
ROWDIR    DW   ?              ;Broj reda tekućeg direktorijuma.
LEVEL     DW   ?              ;Broj kolone tekućeg direktorijuma.
NUMROWS   DW   ?              ;Ukupan broj korišćenih redova matrice.
TREEFLAG  DW   ?              ;Indikator uspeha pri formiranju stabla.
ENDPATH   DW   ?              ;Adresa iza kraja staze DIRPATH
NEWLEVEL  DB   ?              ;Indikator postojanja sledećeg nivoa.
DISK      DW   ?              ;ASCII oznaka diska.
```

Konstante, promenljive i tabela

Možda je lakše odgovoriti na drugo pitanje. Registar AX na izlasku iz funkcije 4EH i 4FH nosi celokupnu informaciju o obavljenom pretraživanju. Ako pretraga nije uspela, sadržaj AX će biti 0012H. U protivnom možemo smatrati da je sve u redu.

Ako je pretraga uspela, kompletno ime poddirektorijuma ili datoteke koju smo tražili, zajedno sa ostalim podacima, biće upisano u poseban blok memorije za komunikaciju sa diskom. Ako nešto sami nismo menjali, biće to uvek deo zaglavlja našeg COM programa (PSP). Nama je interesantno samo ime, koje ćemo naći na adresi 009EH, u ASCIIZ formatu. Ovo važi i za jednu i za drugu funkciju pretraživanja.

Potprogram GETLIST upravo obavlja pretraživanje direktorijuma na opisan način. Prethodno je formirana staza sa *.* e traže se isključivo poddirektorijumi. Ako ih bude bilo, prve dva će obavezno biti „." i „..", što nama ništa ne znači, pa ćemo ih preskočiti. Sve ostale ćemo prikupiti i upisati u listu DIRNAMES. Pri tome, vodimo računa da li u listi ima dovoljno mesta.

NOVI PODDIREKTORIJUMI

Ako je tekući direktorijum pokazao grananje, sledeći korak će biti da se svi ti poddirektorijumi upišu u matricu DIRMAP, vodeći računa o strukturi stabla i pazeći da ne narušimo već postojeće veze. O tome brinu potprogrami: CONNECT, INSROWS i MAKEDIRS.

Pre svega, desno od tekućeg direktorijuma u matrici DIRMAP dopisuje se prvi poddirektorijum (potprogram CONNECT). Ova operacija ne remeti ostatak tabele, tako da ne predstavlja poseban problem. A ukoliko i ne postoji više od jednog poddirektorijuma, posao je obavljen.

Za veći broj poddirektorijuma, mora se obaviti *umetanje* novih redova u tabelu DIRMAP, odmah ispod tekućeg reda. Tu se jasno javlja problem očuvanja svih vertikalnih veza, što je i ključni zadatak potprograma INSROWS.

Konačno, u novoformirana polja upisuju se svi poddirektorijumi i povezuju odgovarajućim vertikalnim vezama, što radi potprogram MAKEDIRS. Odmah iza toga, vrši se skok na sledeći red i ciklus se ponavlja.

ULAZNI I IZLAZNI PARAMETRI

Pri pozivu potprograma FORMTREE, treba upamtiti da na ulazu obavezno moramo definisati ime diska u promenljivoj DISK. Na primer, ako nas interesuje disk A, upisaćemo u DISK odgovarajući ASCII kod 41H i tek onda pozvati FORMTREE:

```
MOV     DISK,41H
CALL    FORMTREE
```

Na izlazu moramo analizirati sadržaj promenljive TREEFLAG. Moguće su tri vrednosti, sa sledećim značenjem:

- **00H** Disk se ne odaziva (preciznije, nije uspelo čitanje sektora 1, staza 0, strana 0).
- **01H** Stablo direktorijuma je uspešno formirano u tabelama DIRMAP i DIRNAMES.
- **02H** Stablo je previše komplikovano (ima više od osam nivoa ili više od 256 redova, ili sva imena nisu stala u listu DIRNAMES).

Sledećeg puta ćemo videti kako formirano stablo možemo prikazati na ekranu i onda birati željeni direktorijum samo pomeranjem kursora.

DOBRO DOŠLI U VERMONT

Vermont Views Designer V2.0 omogućuje interaktivno kreiranje menija i maski za unos podataka i, potom, njihovo prevođenje u C izvorni kod. Program RAC.C i heder PODACI.H nastao je na ovaj način, prevođenjem primera koji je opisan na strani 35. Dobijeni izvorni kod se sada može jednostavno modifikovati prema želji i, naravno, potrebama, i povezivati sa sopstvenim C aplikacijama.

```
/*---------------------------------------------------------------*/
/*  This program was generated with the VVDOCGEN program using    */
/*  Designer library report rac.rpt on Fri May 03 11:34:11 1991*/
/*---------------------------------------------------------------*/

/*---------------------------------------------------------------*/
/*  Include the necessary Vermont Views header files              */
/*---------------------------------------------------------------*/

#include <vv_sys.h>
#include <vv_data.h>
#include <vv_dss.h>
#include <vv_main.h>

/*---------------------------------------------------------------*/
/*  Include the Designer-generated header files that define       */
/*  the structures used by data-entry forms                       */
/*---------------------------------------------------------------*/
#include <podaci.h>

/*---------------------------------------------------------------*/
/*  define the data structures to hold data-entry form data       */
/*---------------------------------------------------------------*/

PODACI podacidata;

/*---------------------------------------------------------------*/
/*  Start of program                                              */
/*---------------------------------------------------------------*/
#ifdef LINT_ARGS
int main(int argc, char *argv[])
#else
int main(argc, argv)
int argc;
char *argv[];
#endif
{

/*---------------------------------------------------------------*/
/*  Define pointer to library structure                           */
/*---------------------------------------------------------------*/

   DLIBPTR dlibp;

/*---------------------------------------------------------------*/
/*  Define pointers to the used data-entry forms                  */
/*---------------------------------------------------------------*/

   DFORMPTR podacidfmp;

/*---------------------------------------------------------------*/
/*  Reference the argument list parameters to quiet some          */
/*  compilers                                                     */
/*---------------------------------------------------------------*/

   argc;
   argv

/*---------------------------------------------------------------*/
/*  Initialize                                                    */
/*---------------------------------------------------------------*/

   init_vv();      /* initialize Vermont Views */

   vs_clr();       /* clear screen */

/*---------------------------------------------------------------*/
/*  open Designer form library                                    */
/*---------------------------------------------------------------*/

   dlibp = dl_open("rac.vvd");

/*---------------------------------------------------------------*/
/*  Load in used data-entry forms                                 */
/*---------------------------------------------------------------*/

   podacidfmp = dl_fmget("podaci", &podacidata, NULLP, NULLP,
   dlibp);

/*---------------------------------------------------------------*/
/*  Close Designer form library                                   */
/*---------------------------------------------------------------*/

   dl_close(dlibp);

/*---------------------------------------------------------------*/
/*  Process used data-entry forms                                 */
/*---------------------------------------------------------------*/

   fm_proc(0, podacidfmp);

/*---------------------------------------------------------------*/
/*  Free used data-entry forms                                    */
/*---------------------------------------------------------------*/

   fm_free(podacidfmp);

/*---------------------------------------------------------------*/
/*  Clean up screen and return to operating system                */
/*---------------------------------------------------------------*/

   vs_clr();                      /* clear screen */

   csr_mv(vs_rowq() - 1, 0);      /* move cursor to bottom */

   exit_vv();                     /* de-initialize Vermont Views */

   return(0);
}

typedef struct                          /* Data form data structure */
{
   UCHAR item0[35];
   UCHAR item1[30];
   UCHAR item2[17];
   UCHAR item3[11];
   UCHAR item4[24];
} PODACI;
```

WINDOWS I TASTATURA

Jednom svojom osobinom *Windows 3.0* nas ponovo vraća u predistorijska vremena kada se *PC Tools*-om, ili kakvom sličnom alatkom, hakerisalo po drajveru za tastaturu.

Jedno od razočarenja koje su doživeli YU korisnici programe *Windows 3.0* je i osobina ovoga okruženje da direktno preuzima kontrolu nad hardverom i zaobilazi rezidentne programe koji pod DOS-om kontrolišu tastaturu. Uobičajene posledica ovakvoge načina reda *Windows*-a su YU znaci razbacani po celoj tastaturi.

Ovo je samo privremeni problem, koji će prestati da postoji sa pojavom nove verzije *Windows*-a sa ugrađenom podrškom za kodnu stranicu CP 852, u kojoj se nalaze znaci latiničnih pisama istočnoevropskih zemalja uključujući i Jugoslaviju. Kada se ova verzija pojavi, instalacija YU fontova i drajvere za tastaturu će se svoditi na podešavanje paramelara u **Control Panel**-u i njegovom **International** servisu. U međuvremenu, možemo na jednostavan način preraditi *Windows*-ov drajver za emeričku (US) tastaturu tako da podržave YUSCII raspored slova.

Windows-i i aplikacije pisane za njih rade pod ANSI setom znakova. ANSI set znakova je u prvih 7 bita (0-127 dekadno) identičan ASCII setu, dok se u 8 bitu (128-255 dec) razlikuje. *Windows*-i koriste ovaj osmi bit da u drugi deo seta upišu znakove specifične za pojedine zemlje. Podešavanje parametera (znaci, sortni redosled i raspored na tastaturi) vrši se iz **Control Panela** izborom imena odgovarajućih drajvera za **language.dll** i **keyboard.drv** u **[boot]**, i **keyboard.dll** i **oemansi.bin** u **[keyboard]** sekciji datoteke SYSTEM.INI.



Slika 1. Sadržaj orgineine detoteke KEYBOARD.DRV



Slika 2. Sadržaj prapravljene detoteke KEYBOARD.DRV

Relat. sector/ Displacement	Stari sadržaj	Novi sadržaj	Opis
1/0505	59 - Y	5A - Z	zamena Y sa Z
2/0016	5A - Z	59 - Y	zamena Z sa Y
9/0286	59 - Y	5A - Z	zamena Y sa Z
9/0308	5A - Z	59 - Y	zamena Z sa Y
10/0083	40 - @	22 - "	dvostruki navod iznad 2
10/0091	5E - ^	27 - '	jednostruki navod iznad 6
10/0099	3C - <	3B - ;	tačka zarez iznad zareza
10/0101	3E - >	3A - :	dvotačka iznad tačke
10/0106	3B - ;	7E - č	tipka desno od L
10/0107	3A - :	5E - Č	-"- -"- sa Shift
10/0108	2F - /	60 - ž	tipka desno od tačke
10/0109	3F - ?	40 - Ž	-"- -"- sa Shift
10/0110	60 - `	3C - <	tipka levo od jedinice
10/0111	7E - ~	3E - >	-"- -"- sa Shift
10/0112	5B - [	7B - š	tipka desno od P
10/0113	7B - {	5B - Š	-"- -"- sa Shift
10/0114	5C - \	2F - /	tipka desno od =
10/0115	7C - |	3F - ?	-"- -"- sa Shift
10/0116	5D -]	7C - đ	druga tipka desno od P
10/0117	7D - }	5C - Đ	-"- -"- sa Shift
10/0118	27 - '	7D - ć	druga tipka desno od L
10/0119	22 - "	5D - Ć	-"- -"- sa Shift
10/0122	5C - \	2F - /	tipka desno od =
10/0123	7C - |	3F - ?	-"- -"- sa Shift

Tabela potrebnih izmena sadržaja datoteke KEYBOARD.DRV

Ukoliko *Windows*-e instaliramo da rade sa američkom tastaturom i američkim jezičkim interfejsom (sa parametrima u **Control Panel**-u **Country: United States, Language: English (American), Keyboard: US**) instališe se samo **keyboard.drv** u **[boot]** sekciji, dok se ostali drajveri ne instališu (polja ostaju prazna). Ovo je standardno jezičko okruženje za *Windows*-e i oni kada rade pod njim ne vrše nikakve dodatne konverzije, iako da se svi podaci vezani za tastaturu nalaze u datoteci KEYBOARD.DRV iz WINDOWS\SYSTEM direktorijuma.

Izmenom sadržaja određenih adresa uz pomoć bilo kojega disk-sektor editora, možemo ovaj drajver prilagoditi YUSCII rasporedu na tastaturi. Prikazivanje YU znakova na ekranu monitora zavisi od ekranskih i printerskih fontova, što predstavlja posebnu temu.

Pre započinjanja intervencija u KEYBOARD.DRV datoteci poželjno je napraviti kopiju orginalnog drajvera (na primer: COPY KEYBOARD.DRV KEYBOARD.OLD), a zatim pokrenuti neki od disk-sektor editora (*PC Tools* itd.) i sa **View/Edit** ući u sadržaj datoteke KEYBOARD.DRV. U datoteci treba izvršiti izmene prikazane u Tabeli. Oznaka relativnog sektora i lokacije dati su decimalno, a sadržaj lokacije je dat u heksadecimalnim vrednostima.

Na slikama 1 i 2 prikazane su kopije ekrana iz *PC Tools*-a za Rel. Sector 10 u kome se vrši najviše izmena. Na slici 1 prikazan je sadržaj orginalnog, a na slici 2 prepravljenog drajvera. Ukoliko opisan raspored slova i simbola ne odgovara u potpunosti iz sheme date u Tabeli, lako se može napraviti raspored po želji, upisivanjem odgovarajućih kodova na pobrojane lokacije.

Zoran Navratil

DVOSTRUKI EKRAN

Postoje situacije u kojima samo jedan monitor nije dovoljan da bi se prikazali rezultati rada neke aplikacije. Pokušaćamo da vam ukažemo na pogodnosti, ali i da istaknemo neke problema vezane za instalaciju drugog monitora.

Pavle Peković

Povezivanje drugog monitora je moguće na dva načina. Jedan je dodavanje nove grafičke kartice pored postojeće, a drugi je korišćenje specijalno dizajniranih grafičkih kartica na kojima se nalazi jedan EGA ili VGA i jedan Hercules kompatibilni adapter. S obzirom da su ove druge kartice novost, te da su retkost na računarskom tržištu, a i cena im je doste velika, ograničićemo se na prvo rešenje – dodavanje još jedne grafičke kartice pored već postojeće.

Jedna grafička kartica mora biti ili MDA ili Hercules, a druga ili EGA ili VGA. Druge kombinacije nisu moguće, što se može veoma jasno uočiti ako pogledamo mapu video memorije (tabela 1).

Dakle, ako imete EGA ili VGA kartu, potrebno je dokupiti MDA ili Hercules kartu (preporučujemo Hercules) i odgovarajući monitor, ili obrnuto. Cela dalja procedura se svodi na to da ubacite novu karticu u odgovarajući slot na matičnoj ploči i da priključite monitor. Eventualno, ako ste imali Hercules, moraćete da izmenite setup, i to onaj deo vezan za video kartice, gde ćete staviti EGA ili VGA.

Pre startovanja sistema, potrebno je uraditi samo još jednu stvar. Ako je vaša EGA ili VGA kartica 16-bitna, morate je podesiti da radi kao 8 bitna iz razloga što jedan 16-bitni adapter u svom 128K adresnom prostoru mora biti sam. Ovo je možda jedina mana cele ideje oko upotrebe dva monitora, jer se upotrebom 16-bitne karte kao 8-bitne smanjuje brzina na otprilike polovinu u nekim video modovima. Ovo će biti veoma neprijatno za one korisnike kojima je brzina video karte veoma važna u njihovom radu sa računarima, međutim, većina korisnika ovo smanjenje brzine jedva da će primetiti. Naravno, treba pomenuti da se smanjenje brzine odnosi samo na 16-bitne EGA i VGA karte, ali ne i na Hercules i MDA, kao i druge 8 bitne karte, kojima je sasvim svejedno da li su same ili ne. Da biste videli koliki su gubici u brzini i u kojim video modovima, napravili smo mali test. Na testu su se našle tri VGA video kartice: Dve 16-bitne Genoa S-VGA i Optima S-VGA, kao i jedna 8-bitna Paradise + VGA. Sve kartice su testirane posebno, kao i zajedno sa jednom Hercules klon kartom. Iz tabele se jasno vidi da su 16-bitne karte u nekim modovima izgubile 50% svoje brzine kada su radile zajedno sa Hercules karticom, dok je 8-bitna Paradise + VGA kartica, ostvarivala približne rezultate.

Tabela 1

Hercules	B0000H	64K
MDA	B0000H	4K
CGA	B8000H	16K
EGA	A0000H	32-128K
	B0000H	
	B8000H	
	VGA	A0000H
32-128K		
	B0000H	
	B8000H	
MCGA	A0000H	64K

Memorijska mapa za različite tipove video kartice

Tabela 2

Genoa 1024×768		Paradise 800×600		Optima 1024×768		
Herc. +		Herc. +		Herc. +		
776	1505	891	893	754	1662	40×25 (text)
694	1424	817	817	746	1648	80×25 (text)
694	1423	817	817	747	1650	80×50 (text)
794	791	1097	1098	781	779	320×200×4 (CGA)
713	707	1065	1066	773	772	640×200×2 (CGA)
712	707	1065	1066	771	773	640×200×16 (EGA)
712	707	1073	1073	776	774	640×350×16 (EGA)
794	791	1097	1098	781	780	320×200×16 (EGA)
712	1501	1064	1064	771	1705	320×200×256 (VGA/MCGA)
712	706	1055	1056	771	771	640×480×16 (VGA)
712 ,	706	1055	1056	770	770	640×480×2 (VGA/MCGA)
1122	3168	909	909	963	963	800×600×2 (S-VGA)
1122	3168	909	909	963	963	800×600×16 (S-VGA)
1122	3168	1150	1151	771	1708	640×480×256 (S-VGA)
634	634	855	855			1024×768×2 (S-VGA)
634	634	855	855			1024×768×16 (S-VGA)
741	746	1039	1040	741	741	360×480×256 (VGA)

Napomena: Kartice su testirane programom VIDSPEED i brojevi izražavaju broj bajtova u milisekundi

Rezultati brzinskih testova sa dve (prva kolona) i jednom video karticom (druga kolona)

Ovde valja ukazati na probleme vezane za upotrebu nekih programa, kao što su ubrzivači ekranskog ispisa. Tipičan primer je program PCKSCRN iz paketa PC KWIK POWER PACK, koji je na većini računara pravio probleme sa Hercules karticom. Sa ostalim programima nismo imali nikakvog problema, verovatno i iz razloga što za rad sistema sa dva monitora nisu potrebni nikakvi dodatni drajveri ili bilo kakvi drugi rezidentni programi.

Prilikom startovanja vašeg računara, nećete primetiti ništa neobično što bi se razlikovalo u odnosu na korišćenje samo jednog monitora. Da biste videli prve efekte, potrebno je da otkucate:

MODE MONO

čime ćete izlaz iz vašeg računara preusmeriti na hercules karticu i monitor. Za povratak izlaza na kolor monitor jednostavno otkucajte:

MODE CO80

Stigli smo do onog, možda najvažnijeg pitanja. Kako se korisno mogu upotrebiti dva monitora? Kao što sam na početku teksta pomenuo, postoji jedan broj programa koji u potpunosti koristi prednosti sistema sa dva monitora, kao npr. *AutoCAD*, *DesqView* i mnogi drugi programi. Primera radi, u *DesqView*-u je, kao što svi verovatno znaju, moguće pokretanje više programa u isto vreme. Sada je moguće i prikazivati rezultate njihovog rada u isto vreme na dva monitora.

Ipak, najveću prednost će osetiti programeri, jer većina popularnih dibagera podržava rad sa dva monitora. Na jednom ekranu ćete videti kod programa, a na drugom rezultat izvršavanja tog programa. Ovakav način ispravljanja programa je ne samo jednostavniji, već i brži. Da bi koristili dva monitora, dva najpopularnija dibagera, *CodeView* i *TurboDebuger* moraju se startovati sa:

CV /2

odnosno

TDEBUG /do

Na kraju, valja istaći i jedan rezidentni program koji će vam, ako se odlučite za sistem sa dva monitora, u mnogome olakšati rad. Ime programa je CTRLALT (sve komande se pozivaju tasterima CTRL i ALT), rezidentan je i kada se instalira zauzima nekih 7K. Nismo imali nikakvih problema sa podizanjem ovog programa u *high* memoriju. Program omogućuje kopiranje sadržaja jednog monitora na drugi u svakom trenutku, zatim „skok“ sa jednog monitora na drugi, kopiranje i prenošenje dela ekrana sa jednog monitora na drugi i još mnogo mnogo sitnica koje će vam olakšati rad. Program je *public domain*, tj. za njegovo korišćenje autor ne traži nikakvu nadoknadu. Program toplo preporučujemo, a možete ga naći na Sezamu.

357 [PJ.PASCAL; PC] Ukoliko iz *Turbo Pascal*-a (a možda i iz nekih drugih jezika) konstrte interapt 25h ili 26h za apsolutno čitanje odnosno pisanje po disku, možete da updanete u velike nevolje. Svaka od ovih procedura, naime, ostavlja na steku po jednu reč viška o čemu *Turbo Pascal* ne vodi računa što proizvodi nepredvidljive rezultate koji se obično svode na blokiranje računara.

Rešenje je dato na slikama 1 i 2 – otkucajte i asemblirajte listing sa slike 1, a zatim u program na *Turbo Pascal*-u ugradite deklaraciju sa slike 2 i normalno pozivajte *diskread* sa očiglednim argumentima. Radi se o implementaciji interapta 25h, „susedni" interapt za pisanje po disku se realizuje na sasvim analogan način [preuzeto sa BIX-a, autori **fred robinson i ajgrant**]

358 [INT.SCANJET, PC] Kako da povežete HP *ScanJet* na računar sa VGA karticom? Ukratko rečeno, prilično teško.

Sve je zvučalo jako jednostavno – izvadili smo karticu iz računara sa *Hercules*-om na koma je sve lepo radilo i prebacili je u VGA sistem. Računar je jako lepo radio (tastatura živa, pali se lampica na disku kada se otuca DIR A: itd) samo što na ekranu nije bilo slike. Konsultacija uputstva nije bila od prevelike koristi pošto je isto očito pisano u danima pre nego što je VGA kartica nastala – ostalo nam je samo da pokušamo da konfigurišemo mikroprekidače kao da imamo EGA-u, ali slike i dalje ni od korova.

Pošto smo *Manifest*-om (iz paketa QEMM) ustanovili da VGA karta „blokira" memoriju do c800, prebacili smo mikroprekidače tako da skaner koristi memoriju od ove adrese (to je trebalo odmah uraditi, ali čovek je uvek optimista) i EUREKA – slika se pojavila. Pošto smo se potrudili da u 386MAX.PRO upišemo EXCLUDE=C800-D000, „prošla" je čak i inicijalizacija drajvera za skaner. Međutim, SJTEST neprekidno prijavljuje da test povezivanja (*connection test*) ne prolazi uz čuvenu poruku da „kontaktiramo najbližeg HP dilera". Iz čistog očajanja prebacili smo mikroprekidače tako da memorijska adresa bude D000 i, začudo, sve odjednom proradi! Naravno, ovo je zahtevalo i ponovnu instalaciju 386MAX-a koji je koristio prostor D000-E000 kao EMS *frame* – uspeli smo da ga smestimo u zonu E000-F000 ali da je tamo već bilo nečega, našli bismo se u **velikoj** nevolji! Sreća da je mašina imala IDE disk interfejs a ne recimo neki sa disk BIOS-om.

Ostalo je još da se konfiguriše softver – aplikacija *Scan Gallery* se može instalirati samo u EGA rezoluciji (uz nju ide „antički" *Windows* 1.1) ali je srećom možemo pokrenuti iz *Windows*-a 3.0 pod uslovom da isti radi u *real* modu (WIN /R). Utisak je sasvim drugačiji (800×600 prema EGA – ne vredi porediti!), jedina je nevolja što po svakom startovanju treba „razvlačiti" prozor koji čini *Image Editor*.

Moguće je da oni koji danas kupe *ScanJet* ili *ScanJet+* dobijaju noviju verziju softvera i novu dokumentaciju i da su, prema tome, pošteđeni većine ovih problema. Nama su, međutim, oduzeli čitavo veče pri čemu nam se nekoliko puta činilo da je stvar jednostavno beznadezna!

359 [DS.MSDOS, PC] U prošlomesečnom tekstu „(DOS) POSLE DESET GODINA" potkrala se jedna nepreciznost. Napisali smo, naime, da DOS 5.0 omogućava formatiranje DSDD disketa na 800K bez ikakvih rezidentnih programa. Pokazalo se, međutim, da ovo **ne** treba zahvaliti DOS-u nego BIOS-u u našem novom računaru – niti DOS 5.0 na ostalim mašinama to omogućava, niti je isto nemoguće na DOS-u 3.30 u saradnji sa novim BIOS-om. I dalje je, dakle, potreban program 800, ali isključivo za formatiranje disketa – docnije operacije sa njima nepogrešivo preuzima DOS 5.0, što je značajna prednost u odnosu na 3.30.

360 [ASEM, PC] Kada smo već kod programa 800 – na slici 3 je asemblerski listing programa 800K. Listing je malo duži nego što je u ovoj rubrici uobičajeno, ali ga objavljujemo jer se iz njega zaista može mnogo toga naučiti, ne samo o komunikaciji sa disketama. Autor ovog PD programa je **Alan D. Jones**, a prilog je preuzet sa BIX-a.

361 [COMM, PC] Ukoliko često modemom komunicirate sa VAX-ovima beogradskog ETF-a (327-311 do 327-318) i ako pri tome koristite (hardverski ili emulirani) MNP protokol za korekciju grešaka, svakako ste primetili problem koji se svodi na povremeno prekidanje do tog trenutka korektne veze. Problem se javlja gotovo uvek kada se ispisuju duže datoteke ili na neki drugi način šalje velika količina teksta sa VAX-a ka vašem PC-ju.

Problem se u većem broju slučajeva može rešiti slanjem XON-a – kada primetite predugu pauzu u pristizanju teksta, jednostavno pritisnite par puta Ctrl Q i sva je šansa da će se sve normalno nastaviti. Reklo bi se da (Pupinov) modem pošalje VAX-u XOFF (Ctrl S) kada ne može da primi više podataka, a zatim iz nekog razloga „zaboravi" da pošalje XON (Ctrl Q) kada ponovo bude spreman. Ctrl Q sa tastature nekako „prođe" do VAX-a i on nastavi da šalje podatke.

Recept se, na žalost, ne može primeniti na prenos datoteka (npr. *ZModem*) ali tada je i pucanje veze znatno ređe [Preuzeto sa Sezama, autor **Aleksandar Damnjenović** (*alexa*)]

362 [BASE.CLIPPER, PC] Naredba PACK je jedna od „Ahilovih peta" *Clipper*-a: ako intenzivno brišete podatke, s vremena na vreme morate izvršavati PACK što troši jako mnogo vremena i obično zahteva neposrednu akciju korisnika. Zato se može pribeći malom „triku" – umesto da brišete slog, zamenite sva njegova polja „praznim vrednostima" (alfanumeričke blankovima, numeričke nulama, datume „praznim datumom" itd). Kada vam zatreba da dodate nov zapis, koristite

```
SEEK SPACE(10)
IF NOT FOUND()
    APPEND BLANK
ENDIF
```

Operacija traje jako kratko i potpuno ukida potrebu za pakovanjem baze podataka [Preuzeto sa Sezama, autor **Dušan Pavkov** (*dulezkule*)]

363 [SAMG; PC] Na slici 4 je dat raspored pinova na standardnom 25-pinskom RS232 konektoru. Zvezdicom su označeni pinovi „bitni" za vezu računar-modem ili čak računar-računar [Preuzeto sa Sezama, autor **Slobodan Petrović** (*feniks*)]

364 [UTIL.4DOS, PC] Zanimljiva osobina 4DOS-a koja „promakne" mnogim korisnicima: Opcijom **/U** obezbeđuje se „podizanje" tranzijentnog dela 4DOS-a u *expanded* memoriju – uz utrošak osnovne DOS memorije od samo 250 bajta imate nešto mnogo bolje od COMMAND.COM-a koji sam za sebe zauzima preko 3K. A stvar radi znatno sigurnije – ne može se više dogoditi da neki program „prebriše" COMMAND.COM sa vrha memorije i tako stvori potrebu za njegovim ponovnim učitavanjem sa diska.

Ovo će naročito obradovati one koji imaju ogroman *environment* koristeći na primer **/E:5000U** oformi se radni prostor DOS-a (u ko-

Listing 1

```
public          DISKREAD
CODE            segment         word public
DISKREAD        proc            near
assume          cs              CODE
assume          ds              DATA
assume          es              nothing
; on entry:
;               BP
; SP ==> near return address
;       offset  of disk buffer
;       segment "   "
;       number of sectors to read
;       starting logical sector number (lo)
;       starting logical sector number (hi)
;       drive number (0=A  1=B  etc
; on exit:
;       AX = function result
;            0000 - function successful
;            other   error result as returned by INT 25H
args            buffer ofs      word
                buffer_seg      word
                num_sects       word
                start_sectlo    word
                start_secthi    word
                drive           byte    2
= parmsize
                push            bp
                mov             bp, sp
                push            ds
                mov             AX, 3055H
                int             21H
                cmp             al  4
                jnz             diskread_1
  DOS 3 x
                mov             al  drive
                mov             dx  start_sectlo
                mov             cx  num_sects
                lds             bx  dword ptr buffer_ofs
                jmp             diskread_2
; DOS 4 x
diskread_1:     mov             al  drive
                mov             cx  0FFFFH
                mov             bx  seg int25_parms
                mov             ds  bx
                mov             bx  offset int25_parms
                mov             dx  start_sectlo
                mov             int25_parms i25_slo  dx
                mov             dx  start_secthi
                mov             int25_parms i25_sechi  dx
                mov             dx  num_sects
                mov             int25_parms i25num_sec  dx
                mov             dx  buffer_ofs
                mov             int25_parms i25_bufo  dx
                mov             dx  buffer_seg
                mov             int25_parms i25_bufs  dx
diskread_2:     int             25H
                jnc             e5          ; fix broken stack
                inc             e5
                jc              diskread_3
                xor             ax  ax
diskread_3:     pop             ds
                pop             bp
                ret             parmsize
DISKREAD        endp
CODE            ends
DATA            segment         word public
i25_par_p       struc
i25_slo         dw              ?
i25_sechi       dw              ?
i25num_sec      dw              ?
i25_bufo        dw              ?
i25_bufs        dw              ?
i25_par_p       ends
int25_parms     i25_par_p   <>
DATA    ends
end
```

Asemblerska implementacija interapta 25H i 26H sa *TurboPascal*

Slika 2

```
{$L DISKREAD}
function diskread(drive : byte; start sec : longint;
                  num secs : word; var buffer; ...) : word; external;
```

Funkcija *diskread*

Listing 3

```
[illegible]
```

Listing se nastavlja na sledećoj strani

Nastavak sa prethodne strane

```
                MOV     BYTE PTR DS:[BX] 0SH
                MOV     DX 03F2H
                MOV     AL 10H
                ROL     AL 1
                ROL     AL 1
                AND     AL 03H
                OUT     DX AL
                MOV     CYL 0
                MOV     HEAD 0
;set up sector addresses
FMLOOP:         MOV     SI,OFFSET SA
                MOV     CX SECPERTRK
SASET:          MOV     AL CYL
                MOV     BYTE PTR [SI] AL
                MOV     AL HEAD
                MOV     BYTE PTR [SI+1] AL
                ADD     SI 4
                LOOP    SASET
;display status message
                MOV     AL CYL
                MOV     SI,OFFSET MSG3A
                CALL    STR
                MOV     AL HEAD
                MOV     SI,OFFSET MSG3B
                CALL    STR
                MOV     DX,OFFSET MSG3
                MOV     AH,9
                INT     21H
;format a track
                MOV     RETRY_COUNT 5
DISKCLI:        MOV     DL DRIVE
                MOV     DH HEAD
                MOV     CL 1
                MOV     CH CYL
                MOV     BX OFFSET SA
                PUSH    DS
                POP     ES
                MOV     AL SECPERTRK
                MOV     AH 5
                CLD
                CALL    HEAD
                INT     13H
                JNC     VFYTRK
DEC_RETRY:      MOV     DL DRIVE
                MOV     AH 0
                CALL    INRD
                INT     13H
                DEC     RETRY_COUNT
                CMP     RETRY_COUNT 0
                JNE     DISKCLI
                JMP     FMTERR
;verify the track
VFYTRK:         MOV     DL DRIVE
                MOV     DH HEAD
                MOV     CL 1
                MOV     CH CYL
                MOV     BX OFFSET TRKBUF
                MOV     AL SECPERTRK
                MOV     AH 2
                CLD
                CALL    HOOK
                INT     13H
                JC      DEC_RETRY
                JMP     INCTRK
;bad track - flag clusters in FAT image
;display message
                MOV     DX,OFFSET MSG4
                MOV     AH,9
                INT     21H
;read sectors one at a time to find bad ones
                MOV     SECT,1
READSINGLE:     MOV     DL,DRIVE
                MOV     DH,HEAD
                MOV     CL,SECT
                MOV     CH,CYL
                MOV     BX,OFFSET TRKBUF
                MOV     AL,1
                MOV     AH,2
                CLD
                CALL    HOOK
                INT     13H
                JNC     NXSECT
;compute sector number
                MOV     AL,CYL
                MOV     CL,SECPERTRK*2
                MUL     CL
                MOV     CL SECT
                DEC     CL
                SUB     CH,CH
                ADD     AX,CX
                CMP     HEAD,0
                JE      CCN
                ADD     AX,SECPERTRK
;compute cluster number
CCN:            SUB     AX,RESERVDSEC
                SUB     AX,NUMOFFATS*SECPERFAT
                SUB     AX,(ROOTENTRIES*32)/BYTESPERSEC
                SUB     DX,DX
                MOV     CX,SECPERCLUS
                DIV     CX
                ADD     AX,2
;compute offset into table
                MOV     CL,1
                AND     CL,AL
                SHL     CL,1
                SHL     CL,1
                MOV     DX,AX
                SHR     AX,1
                ADD     AX,DX
;write FF7 +
                MOV     BX OFFSET FAT
                ADD     BX,AX
                MOV     AX 0FF7H
                SHL     AX,CL
                OR      [BX] AX
;next sector on track
NXSECT          INC     SECT
                CMP     SECT SECPERTRK
                JBE     READSINGLE
;increment to next track
INCTRK:         CMP     BREAK 1
                JNE     INCHEAD
                JMP     BREAKEXIT
INCHEAD:        INC     HEAD
                CMP     HEAD 1
                JA      INCTRK2
                JMP     FMLOOP
INCTRK2:        MOV     HEAD 0
                INC     CYL
                CMP     CYL CYLPERDSK
                JNB     SENDLF
                JMP     FMLOOP
;display CRLF
SENDLF          MOV     DX OFFSET CRLF
                MOV     AH 9
                INT     21H
;plug in correct media parameters
                MOV     BX OFFSET BOOTIMG
                MOV     WORD PTR [BX+11],BYTESPERSEC
                MOV     BYTE PTR [BX+13],SECPERCLUS
                MOV     WORD PTR [BX+14],RESERVDSEC
                MOV     BYTE PTR [BX+16],NUMOFFATS
                MOV     WORD PTR [BX+17],ROOTENTRIES
                MOV     WORD PTR [BX+19],SECPERDISK
                MOV     BYTE PTR [BX+21],DESCRIPTOR
                MOV     WORD PTR [BX+22],SECPERFAT
                MOV     WORD PTR [BX+24],SECPERTRK
                MOV     WORD PTR [BX+26],NUMOFHEADS
                MOV     WORD PTR [BX+28],HIDDENSEC
;write diskette boot sector
                PUSH    DS
                POP     ES
                MOV     CYL 0
                MOV     HEAD 0
                MOV     SECT 1
                MOV     BX OFFSET BOOTIMG
                CALL    WRITESEC
                JC      FATALERR
                CMP     BREAK 1
                JE      BREAKEXIT
;write first copy of FAT
                MOV     SECT 2
                MOV     BX,OFFSET FAT
                CALL    WRITESEC
                JC      FATALERR
                ADD     BX 512
                MOV     SECT 3
                CALL    WRITESEC
                JC      FATALERR
                ADD     BX 512
                MOV     SECT 4
                CALL    WRITESEC
                JC      FATALERR
                CMP     BREAK 1
                JE      BREAKEXIT
;write second copy of FAT
                MOV     SECT 5
                MOV     BX,OFFSET FAT
                CALL    WRITESEC
                JC      FATALERR
                ADD     BX,512
                MOV     SECT 6
                CALL    WRITESEC
                JC      FATALERR
                ADD     BX,512
                MOV     SECT 7
                CALL    WRITESEC
                JC      FATALERR
                CMP     BREAK 1
                JE      BREAKEXIT
;display "ok" msg
                MOV     DX,OFFSET MSG1
                MOV     AH,9
                INT     21H
                JMP     RESTORE
;error
FATALERR:       MOV     DX,OFFSET MSG2
                MOV     AH,9
                INT     21H
                JMP     RESTORE
;control-break exit
BREAKEXIT:      MOV     DX,OFFSET MSG5
                MOV     AH,9
                INT     21H
;restore BIOS parameters + INT 23H
; and exit
RESTORE:        SUB     AX,AX
                MOV     DS,AX
                MOV     BX,78H
                MOV     AX,WORD PTR OLD_DISKBASE
                MOV     DS:[BX],AX
                MOV     AX,WORD PTR OLD_DISKBASE+2
                MOV     DS:[BX+2],AX
                LDS     DX,OLD23
                MOV     AX,2523H
                INT     21H
                MOV     AX,4C00H
                INT     21H
TRKBUF          LABEL   BYTE
CODE            ENDS
                END     START
```

Izvorni kod programa *800 K*

me se čuvaju SET promenljive itd) od čitavih 5 kilobajta, ali u *expanded* memoriji! [Preuzeto sa Sezama, autor **Ivica Vujanić** (*ivujanic*)].

364 [UTIL 4DOS, PC]: Toliko što se tiče komplimenata 4DOS-u, da pomenemo i jedan nedostatak. Iz za sada nepoznatih razloga, 4DOS odbija da sarađuje sa komandom *pushd* (direktorijum na stek) iz *Norton Commander*-a i to odbija na prilično čudan način – *pushd* vas zapravo prebaci u željeni direktorijum ali na odgovarajući stek ne stavi baš ništa. Da bi stvar bila još zagonetnija, komanda **radi** ako je izvršite iz BATch datoteke. [Preuzeto sa Sezama, autor **Ivica Vujanić** (*ivujanic*)].

365 [ZAST; PC]: Ukoliko vam neko ponudi SCANV78.ZIP, ne nasedajte provokaciji – ne radi se o novoj verziji poznatog virus detektora firme *McAfee Associates* nego o potencijalno opasnom virusu. Vredi napomenuti da je (za sada) nepoznati zločinac uložio veliki trud u pripremu ovog falsifikata, potpuno je imitirao *McAfee* listu datoteka, ponudio „autentičan" program za verifikaciju i čak uspeo da proizvede ZIP datoteku koja prolazi AV test – pri raspakivanju se dobija poruka *Authentic files Verified! #TJB859 Zip Source: McAFEE ASSOCIATES* koja je veoma nalik na pravu s tim što je serijski broj pogrešan – *McAfee*-ju pripada broj NWM405. To praktično znači da je zaštita koju je PKZIP 1.1 do skora nudio svojim registrovanim vlasnicima „pala" i kao takva postala bespredmetna. [Preuzeto sa InterNet-a, autor **Aryeh Goretsky** (*aryehg@darkside.com*)]

Priloge za ovu rubriku šaljite na adresu „Računari" (za „Bajtove lične prirode"), Bulevar vojvode Mišića 17, Beograd ili preko Sezama u vidu lične poruke uredniku (*mail write dejan*).

ispravka

U prošlom broju sam preuzeo nezahvalan zadatak da pokušam da objasnim suštinu indirektnog adresiranja u C-u (pointeri), ali me je pogled na tekst kada su *Računari* izašli iz štampe malo razočarao u moguće domete mog objašnjenja – ono što je delovalo dosta pregledno na ekranu mog računara, u štampi je dobilo formu koja se teško prati. Napravljena je i jedna omaška – pre deklaracija na kraju teksta

main(int argc, char *argv[])

greškom je izostavljeno

main(int argc, char **argv)

na šta se inače odnosila prva recenica „Ako vam je ova deklaracija pomalo neobična ..." Nadam se ipak da je tekst i pored svega bio koristan.

strncpy

U jednom od prethodnih tekstova sam ukazao na neobičnost i opasnost po tok programa koju krije funkcija **strncpy**. Kopirajući zadati broj elemenata nekog stringa, ona *ne* stavlja terminator na kraju ukoliko string koji se kopira ima više od zadatog broja elemenata, pa na kraju niza „isplivava" slučajan ili prethodni sadržaj. Na moju molbu da proverite šta se dešava u *Turbo C*-u javio se *Luka Frelih* pismom u kome me obaveštava da je ovakvo ponašanje **strncpy** ne samo dokumentovano već i potpuno standardno. Ugrađeni help u TC-u čak upozorava na situaciju koju sam pomenuo – doslovce stoji „odredišni niz može biti neterminisan nulom ukoliko u izvornom stringu ima više od zadatog broja elemenata". Objašnjenje takođe otkriva i drugu osobinu ove funkcije – odredišni niz će biti *popunjen nulama* ukoliko izvorni niz ima manje od zadatog broja elemenata. Provera je pokazala da i MSC verzija ove funkcije radi na identičan način. Jedino što nam još ostaje je da pokušamo da dokučimo koji je smisao ovakvog ponašanja funkcije – očigledno je da neki smisao postoji, ali primenjivost ovih osobina ne deluje očigledno?

promenjiv broj parametara

Jedna od najinteresantnijih osobina C-a je mogućnost kodiranja funkcija koje dozvoljavaju prenos promenjivog broja parametara. Ova osobina se maksimalno koristi – verovatno da nema programa u kome se bar jednom ne pojavi funkcija ovog tipa – **printf**. Kako se pišu ovakve funkcije? Evo jednog primera

```
#include <stdarg.h>
long calc(char *oper, ...)
{
   va_list arg;
   long stck[32];
   int stckptr = -1;
   va_start(arg, oper);
   while (*oper) {
   switch (*oper) {
   case 'x':
   stck[ ++stckptr ] = va_arg( arg, long );
       break;
   case '+':
   stck[ stckptr - 1] += stck[ stckptr ];
       --stckptr;
       break;
...
   }
   ++oper;
}
return( stck[ stckptr ] );
}
/* usage */
printf("Rezultat = %ld\n",
calc("xx+",10L,31L));
```

Funkcija **calc** predstavlja osnovu jednog RPN kalkulatora. Ideja se sastoji u tome da se jednim stringom definišu sve potrebne operacije, dok se sami operandi prosleđuju kao proizvoljan broj parametara funkcije **calc**. Način upotrebe možete videti iz datog primera sabiranja dva broja.

Tri su elementa bitna. Prvi se krije u samoj definiciji funkcije – pošto nismo u stanju da unapred znamo ni tip ni broj parametara **calc** funkcije, upotrebili smo znak **...**. Njegova upotreba nema nikakav dublji smisao već samo naznačava kompajleru da funkcija može imati još parametara o kojima se ništa unapred ne zna, te nema potrebe da vrši proveru njihovog postojanja ili tipa. Drugi element je tip promenjive **arg** – **va_list**. Ovaj tip koji inače u sebi krije pointer je uveden zbog kompatibilnosti – na PC-u se radi o **char ***. I konačno, treći element su dve funkcije (tačnije, dva makroa definisana u **stdarg.h**) **va_start** i **va_arg**. Prvim makroom postavljamo vrednost našeg pointera (**arg**) na prvi parametar u listi, dok drugim uzimamo redom jedan po jedan parametar, navodeći njegov tip.

Da biste do kraja shvatili kako ovaj mehanizam radi, podsetiću da se parametri prenose u funkciju po vrednosti i to slaganjem sdesna ulevo na stek pre poziva funkcije, dok se po ulasku u samu funkciju inicijalizuje poseban registar koji pokazuje na početni parametar (na onaj koji je poslednji gurnut na stek, odnosno onaj koji je prvi u listi parametara funkcije). Ovaj mehanizam radi potpuno nezavisno do same funkcije – napišete li, recimo, u programu **demo(1,2,3,4,5,6)**, svih šest brojeva će biti gurnuto na stek čak i ako je **demo** deklarisan kao **demo(void)** (naravno, kompajler će upozoriti na ovu situaciju, ali će program biti preveden do kraja). Dakle, kada program stigne u našu **calc** funkciju, na steku će se naći niz parametara čiji ni tip ni broj ne znamo. Makroom **va_start** ćemo u pointer **arg** prebaciti adresu prvog parametra i na dalje moramo da se snalazimo sami.

Pretpostavimo da je prvi parametar neki **int**. Ukoliko unutar funkcija stoji, recimo, **i = va_arg(arg, int)**, u promenjivoj **i** će se naći njegove vrednost, a vrednost pointera **arg** će biti povećana za **sizeof(int)** bajtova, ukazujući tako na sledeći parametar. Očigledno je da nas ništa ne sprečava da umesto **int** pokušamo da sa steka „skinemo" neku **float** vrednost ili da nastavimo sa skidanjem parametara i onda kada su svi realno postojeći iscrpljeni – program nema način da proveri ovakve situacije i zato mora postojati neki mehanizam koji će obezbediti da ipak „znamo šta radimo".

Ključ je, bar što se naše funkcije **calc** tiče, u prvom parametru koji smo nazvali **oper**. Njegovim sadržajem stavljamo do znanja funkciji šta da očekuje kao sledeć parametar – kada primeti u stringu znak **x**, program sa steka skida jednu **long** vrednost ... pa šta bude. Ako je niz parametara u saglasnosti sa podacima koje smo zadali stringom **oper**, funkcija će raditi korektno. Ukoliko pogrešimo, dobijaćemo prosto pogrešne rezultate bez ikakve poruke o grešci. Ali ovo je već stara priča u C-u – velika fleksibilnost se mora nečim platiti – jedan deo cene je potreba za povećanom koncentracijom u kodiranju.

Kada jednom počnete da koristite funkcije sa promenjivim brojem parametara, shvatićete koliko su često elegantan način za rešavanje problema. Recimo, jedna od njih koji me je dugo mučio u mojim „pascal danima" je kako konstruisati neku univerzalnu rutinu za ispis greške – poruka često treba da sadrži ne samo tekst već i parametre koji su bilo brojevi bilo stringovi. U C-u se to rešava izuzetno lako

```
void error(int err_code, ...)
{
   va_list arg;
   va_start(arg, err_code);
   switch (err_code) {
   case FILE_NOT_FOUND:
       printf("Datoteka %s ne postoji!\n",
       va_arg(arg, char *));
       break;
   case DISK_SPACE:
       printf("Prostor na disku (%ld bytes)
       nedovoljan.\n",va_arg(arg,long));
       break;
...
   }
}
/* usage */
error(FILE_NOT_FOUND,filename);
error(DISK_SPACE,disk_free);
```

Za kraj, samo da napomenem i jednu očiglednu osobinu sistema, a to je da se parametri mogu uzimati samo sekvencijalno. U ovom poslednjem slučaju funkcije **error** u slučajevima kada ima više od jednog parametra treba biti posebno obazriv da se ne napiše nešto poput

printf("Datoteka %s veća od %s", va_arg(arg,char*),va_arg(arg,char*));

Iz diskusije u jednom od ranijih nastavaka proizilazi da se ne može znati redosled kojim će biti izvedena dva makroa, pa se prvi naziv može pojaviti i levo i desno od teksta „veća od", što poruku unapred čini besmislenom. Zato je potrebno parametre izvaditi van **printf** funkcije.

#pragma

Službeno reči koje započinju znakom **#** ne spadaju u sam C jezik, ali su za njega čvrsto vezani. Radi se o direktivama samom C prevodiocu, bilo njegovom pretprocesorskom delu, bilo samom kompajleru. Na svakodnevni **#include** ste već navikli, ali je jedna od direktiva prilično zanemarena, iako pruža dosta pogodnosti. Direktiva **#pragma** je namenjena kompajleru, kome treba da ukaže na zahteve koji su „mašinski zavisni" i samim tim specifični za pojedine tipove računara. Ovu namenu treba uslovno shvatiti – radi se, zapravo, o nekoj vrsti „otvorenih vrata" ka prostoru u kojoj je svakom proizvođaču C kompajlera dozvoljeno da ostavlja svoje dodatke.

Naravno, osnovna namena ove direktive se uvek poštuje, tako da ona može biti i ignorisana a da to nema značajnih posledica po generisani program. Odatle proizilazi i specifičan način njene obrade – ukoliko je nepoznata, biće ignorisana uz eventualno upozorenje.

Kod Microsoft C-a pragma je prilično popunjena – od standardnih parametara listinga do dubokog zadiranja u način generisanja i organizacije koda. Konkretno, pragma **optimize** omogućava da utičemo lokalno na optimizaciju programa, tako što navodimo **on** ili **off** stanje

#pragma optimize ("l" ,off)

Slova koja se mogu naći unutar znakova navoda su ista ona koja idu uz **O** prekidač u komandnoj liniji. Slično je i sa

#pragma check_stack (off)

koja lokalno isključuje generisanje koda za proveru stanja steka. Ide se i dalje od ovoga

#pragma alloc_text(CODESEG_A, f1, f2)

nalaže kompajleru da funkcije **f1** i **f2** rasporedi u isti segment **CODESEG_A**, pa čak i

#pragma same_seg(var1, var2)

čime se optimizacionom algoritmu daju šire mogućnosti, stavljajući mu do znanja da se promenjive **var1** i **var2** nalaze u istom segmentu.

Upotreba **pragma** direktive ima puno prednosti nad korišćenjem kompajler prekidača iz komandne linije – na kod se može delovati lokalno, dokumentacija je potpunija, svako ko kompajlira program istim kompajlerom dobiće potpuno isti rezultat kao i vi bez potrebe da mu se posebno dokumentuje način kompajliranja, a za one koji ne koriste isti kompajler opcije će prosto biti ignorisane.

TEŽAK (MAŠINSKI) POREDAK

Ma šta se mislilo, umemo mi da damo i tešku Pitalicu. Ovoga puta veoma uspešnu – sortiranja brojeva na hipersiromašnom procesoru zahtevalo je pravu programersku virtuoznost. Pa ipak, problem je rešiv – naši čitaoci su uspeli da u raspoloživa 32 bajta memorije sortiraju čak šest podataka!

Podsetićemo se, pre svega, same pitalice. Na raspolaganje smo vam stavili rudimentarni mikroprocesor koji pristupa memoriji od čitava 32 bajta (svaki bajt osam bita, *mem[0]*...*mem[31]*), ima osmobitni akumulator (*acc*) i petobitni brojač naredbi (*pc*). Mikroprocesor podržava isključivo apsolutno adresiranje, tj. u okviru svake instrukcije navodi se oznaka memorijske ćelije u koju je upisan argument.

Svaka instrukcija se kodira u jednu memorijsku ćeliju (8 bita) pri čemu tri najstarija bita (5-7) predstavljaju kod naredbe u skladu sa tabelom 1 dok preostalih pet bita (0-4) predstavljaju adresu operanda. Ukoliko je instrukcija takva da po svojoj prirodi nema argument (recimo HLT), adresni bitovi se ne konstatuju tj. njihov sadržaj nije bitan.

Kratak opis instrukcija jedva da je potreban. LDA i STA prenose podatke iz memorije u akumulator i natrag (primetimo da postoji samo apsolutno adresiranje, tj. ako nam je potreban broj 1, moramo ga imati negde u memoriji da bismo ga doveli u akumulator). JMI je uslovni skok na zadatu lokaciju ako je broj u akumulatoru negativan tj. ako mu je najviši bit setovan (procesor, dakle, nema nikakve liego-modifikovanje većeg broja naredbi u vej). ADD je sabiranje sadržaja akumulatora sa sadržajem memorijske lokacije *mem[n]* pri čemu se rezultat upisuje u akumulator a eventualno prekoračenje ignoriše. CPL promena znaka broja u akumulatoru (brojevi se memorišu u potpunom komplementu tj. komplementiranjem broja 01101100 se, na primer, dobija 10010100) uz ignorisanje eventualnog prekoračenja. HLT je kraj rada a NOP naredba bez dejstva.

Pre početka rada u memoriju se upisuje program i potrebne konstante odnosno podaci (iste ćelije se, naravno, mogu tretirati jednom kao program a drugi put kao podaci, pošto je „samo-modifikacija" programa jedini način da se ostvari indirektno adresiranje) a zatim i sadržaj registra *pc* što praktično znači da program ne mora počinjati od lokacije *mem[0]*. Izvršavanje započinje i traje do nailaska na naredbu HLT – ako brojač naredbi u nekom trenutku „pređe" preko ćelije *mem[31]* izvršavanje se nastavlja od instrukcije upisane u *mem[0]*.

Pošto smo upoznali procesor, da se podsetimo i probleme – trebalo je napisati program koji sortira što više podataka u rastući (ili, ako vam više odgovara, opadajući) poredak. Brojevi su, u skladu sa potrebama programera, upisani u sukcesivne memorijske ćelije i tretiraju se kao **označeni** tj. oni brojevi čiji je najviši bit setovan su negativni u potpunom komplementu. Po izvršavanju programa u iste ćelije trebalo je da budu upisani isti brojevi ali odgovarajućim redosledom!

Za razliku od prethodne (kada smo računari vrednost polinoma) odziv na ovu mikroprocesorsku Pitalicu je bio izrazito slab – primili smo svega 7 pisama i dva odgovora preko Sezama. Svi rešavači su konstatovali da je problem (pre)težak, a mnogi su se pitali da li se uopšte i može rešiti osim u trivijalnom slučaju sortiranja dva broja.

Na svu sreću, rešenje postoji i donesi prvu nagradu (1400 dinara) *Vladanu Vučkoviću* iz Niša. On je napisao program sa slike 2 koji sortira čak šest brojeva upisanih u ćelije *mem[1]*, *mem[2]*... *mem[6]* u opadajući poredak; izvršavanje programa počinje od adrese 5. U čemu je tajna ovog programa? Pre svega u tome što je autor pretvarao jednu instrukciju u drugu: kada se na ADD doda 00100000, dobija se STA a sledećim dodavanjem broja 01000000 iz STA se dobija LDA. Time je automatski ušteđena gomila koda za petljama. Druga majstorija je izlazni kriterijum – ako naredbu LDA posmatramo kao broj, ona je negativna (100xxxxx). Međutim, kako postepeno smanjujemo adresu i pređemo preko LDA 0, dobijamo novu naredbu JMI čiji bi kod bio 011xxxxx i koja je, ako je posmatramo kao broj, pozitivna. Na taj način na izlasku iz programa štedimo jedno sabiranje i jedno poređenje a dakako i odgovarajuću konstantu koju bismo morali da čuvamo negde u memoriji.

Napominjemo, međutim, da 6 nije konačni domet ovog mikroprocesora – iz gimnazijskih dana urednika ove rubrike „zaostali" su programi koji sortiraju 8 i 9 podataka. Testiranje na simulatoru (u vreme kada smo se ovim bavili mogli smo samo sanjati o nekom brzom i komfornom simulatoru NAR-a 1 koji bi radio „kod kuće") pokazalo je da rešenje za 9 brojeva ima jednu greščicu (ko zna, možda bi se i delo sredi?) a ono za 8 funkcioniše besprekorno. Ovde ga, međutim, nećemo objaviti – ostavljamo vam „domaći zadatak" da ga samostalno pronađete ili da izmislite nešto još bolje.

Druga i treća nagrada putuju u istu kuću – zaslužili su ih *Josip* i *Zdravko Krstulović* iz Zagreba (1000 odnosno 700 dinara) koji su poslali dva suštinski različita programa koji sortiraju pet podataka. Na žalost, ova dva čitaoca nam nisu dostavile adresu (ili je ona bila na kovertu koji odvajamo od priloga) pa ih molimo da propust isprave kako bismo im poslali nagradu. Od preostalih 6 rešenja, jedno je nudilo sortiranje četiri broja (*Siniša Karamarković* iz Tuzle), dva sortiranja tri broja (*Snežana Banić* iz Beograda i *Robert Paušić* iz Kikinda), dva sortiranje svega dva podatka (što, kao trivijalno rešenje, nije bodovano za godišnje takmičenje) a poslednje je bilo pogrešno. Više sreće sledeći put!

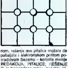
Slika 2

```
[illegible]                         ; početak lokacije
[illegible]                         ; podatak 1
[illegible]                         ; podatak 2
[illegible]                         ; podatak 3
[illegible]                         ; podatak 4
[illegible]                         ; podatak 5
[illegible]                         ; podatak 6
[illegible]                         ; konstanta
[illegible]                         ; konstanta
[illegible]                         ; <--- ulazna tačka
[illegible]
```

Program *Vladana Vučkovića* iz Niša sortira šest brojeva upisanih u ćelije *mem[1]*... *mem[6]* u opadajući poredak

NAJMANJI OD NAJVEĆIH

Svi celi brojevi između 1 i 16 su upisani u polja na slici, jedno polje u jednom kvadratiću. Zatim je u svaki kružić upisan zbir podataka sa četiri polja koja ga okružuju. Dobijenih 9 zbirova su sortirani po veličini, a onda su odbačena dva najveća. Rasporedite brojeve tako da sledeći (treći) **najveći** zbir bude **što manji**.



Rešenja pitalice, zajedno sa **obrazloženim** programima koje ste napisali, pošaljite na uobičajenu adresu: „Računari" (za Dejanove pitalice), Bulevar vojvode Mišića 17, Beograd tako da prispu **pre** 10. juna 1991. Sva pisma sa korektnim rešenjima konkurišu za novčane nagrade od 1400, 1000 i 700 dinara dok će kuponi (ili njihove fotokopije) na koje je upisan identifikacioni broj učestvovati u godišnjem takmičenju rešavača Pitalica. Identifikacioni broj dobijate tako što u prve tri kućice upišete **poslednje** tri cifre nečijeg broja telefona, u sledeće dve godinu vašeg rođenja i na kraj dodate dve cifre po izboru. Obratite pažnju da sva rešenja koja šaljete u toku godine nose isti identifikacioni broj. Osim običnom, rešenje ove pitalice možete da pošaljete i elektronskom poštom posredstvom Sezama – koristite menije REDAKCIJA, PITALICE, REŠENJE ili, iz komandnog moda, pošaljite ličnu poruku korisniku pitalice (*mail write pitalice*). Sva rešenja pristigla elektronskom poštom ravnopravno učestvuju u izvlačenju nagrade i konkurišu za **specijalnu nagradu** od 1000 dinara koja će biti dodeljena najboljem rešenju koje je dostavljeno modemom!

Slika 1

Kod	Mnemonik	Opis
000	NOP	Bez dejstva
001	ADD n	acc+mem[n] --->> acc
010	STA n	acc --->> mem[n]
011	JMI n	n --->> pc, ako je acc<0
100	LDA n	mem[n] --->> acc
101	CPL	-acc --->> acc
110	HLT	Kraj rada
111	NOP	Bez dejstva

Rudimentarni mikroprocesor sa akumulator, petobitnim brojačem naredbi i trideset i dva bajta memorije

```
66: Najveći od preostalih sedam zbirova je ____________

Ime i prezime     ______________________________
Adresa            ______________________________
Mesto             ______________________________
Kompjuter         ______________   Vreme ________

Identifikacioni broj:  |_| |_| |_|  |_| |_|  |_| |_|
                       iz broja tel.  god rođ.  po izboru
```

PC AT 286, 386-SX, 386
ISPORUKA ODMAH

MODEL	SC 2016	SC 2025!!!	SC 3020 SX	SC 3033 CACHE 64 Kb	SC 4033
PROCESOR	INTEL 286	INTEL 286	INT 386 SX	INTEL 386	INTEL 486
CHIP SET	HEADLAND	HEADLAND	HEADLAND	C&T	OPTI
RADNI TAKT	16 MHz	25 MHz!!!	20 MHz	33 MHz	33 MHz
LAND TEST	21 MHz	33 MHz!!!	26 MHz	58 MHz	150 MHz
KUĆIŠTE	BABY AT	BABY AT	BABY AT	BABY AT	BABY AT
NAPAJANJE	200 W	200 W	200 W	200 W	220 W
RAM	1 Mb	2 Mb	4 Mb	4 Mb	4Mb
MAX RAM	5 Mb	8 Mb	8 Mb	16 Mb	16 Mb
FLOPPY	1.2 Mb	1.2 Mb	1.2 Mb	1.2 Mb	1.2 Mb
HARD DISK	ST 157 A 44 Mb	ST 157 A 44 Mb	ST 157 A 44 Mb	ST 157 A 44 Mb	ST 157 A 44 Mb
HD FD CONT	AT BUS INT 1 1	AT BUS INT 1 1	AT BUS INT 1 1	AT BUS INT 1 1	AT BUS INT 1 1
CORETEST	T/T 1 ms 740 K/S	T/T 1 ms 740 K/S	T/T 1 ms 740 K/S	T/T 1 ms 740 K/S	T/T 1 ms 740 K/S
SER INT	1 RS 232	1 RS 232	1 RS 232	2 RS 232	2 RS 232
PAR INT	2 CENTR	2 CENTR	2 CENTR	2 CENTR	2 CENTR
TASTATURA	101 ALPS	101 ALPS	101 ALPS	101 ALPS	101 ALPS
GRAF KART	HERCULES	HERCULES	HERCULES	HERCULES	HERCULES
MONITOR	AMBER 14"	AMBER 14"	AMBER 14"	AMBER 14"	AMBER 14"
CENA DIN.	**30000,00**	**38000,00**	**48000,00**	**66000,00**	**108000,00**

PREKO 70% UGRAĐENIH DELOVA SU IZ USA ILI JAPANA

16" COLOR MONITOR EIZO 9070S (1200*800) I VGA CARD 1 MB 1024*768	39000,00 DIN	*	
14 " VGA COLOR MONITOR I VGA CARD 1 Mb 1024*768	16000,00 DIN	*	(ZVEZDICOM SU OZNAČENE DOPLATE U ODNOSU NA OSNOVNE KONFIGURACIJE)
HD 90 Mb ST 1102 A IDE 16 ms	8000,00 DIN	*	
HD 130 Mb ST 1144 A IDE 18 ms	12000,00 DIN	*	
HD 210 Mb CONNER IDE 16 ms	26000,00 DIN	*	
HD 210 Mb CDC IDE 15 ms	26000,00 DIN	*	
TOWER KUĆIŠTE + 220 W	3500,00 DIN	*	
CYREX 387—33 MHz KOPROCESOR	21000,00 DIN		(100—300% BRŽI OD INTELA 387)
IIT 287—10 MHz KOPROCESOR	6000,00 DIN		(100—300% BRŽI OD INTELA 287)
IIT 287—20 MHz KOPROCESOR	9000,00 DIN		(100—300% BRŽI OD INTELA 287)
FLOPPY DISK 1.44 Mb TEAC	3500,00 DIN		
MIŠ GM 6000 (RS 232)	1600,00 DIN		
EPSON LX 400 (A4 9 PINA)	11000,00 DIN		(PRINTERI SU SA UGRAĐENIM YU ZNAKOVIMA ŠŽČĆĐ I KABLOM ZA VEZU SA RAČUNAROM)
EPSON LQ 550 (A4 24 PINA)	21000,00 DIN		
EPSON LQ 1010 (A3 24 PINA)	25000,00 DIN		
EPSON LQ 1050 (A3 24 PINA)	35000,00 DIN		
EPSON FX 1050 (A3 9 PINA)	24000,00 DIN		
EPSON LX 1050 (A3 9 PINA)	20000,00 DIN		
RAM CARD 2 Mb (LASER JET)	9000,00 DIN		

CENE SU PODLOŽNE PROMENAMA ZBOG PROMENE KURSA DOLARA.
NAČIN PLAĆANJA: 100% AVANS — ISPORUKA ODMAH PO UPLATI.

PREDUZEĆE „ŠUTLIĆ"

KOSOVSKA 32/1. SPRAT, 11000 BEOGRAD,
RADNO VREME: 9—17 h, SUBOTOM 10—14 h
ZVATI OD 9—13 h: TEL. 011/699-248

GARANCIJA 12 MESECI, SERVIS I ODRŽAVANJE
BEOGRAD 04. 06. '91.

DATE CENE OPREME VAŽE ZA PRAVNA I FIZIČKA LICA.
(OD 01. 02. '91. PLAĆA SE SAVEZNI POREZ NA PROMET 3%)

GAMA Electronics Beograd
Mišarska 11
Tel: 011/332-275, Fax: 011/335 902
Radno vreme: pon.-petak **9-17**
GAMA Electronics Zagreb
Tel.+Fax: 041/326-841
GAMA Electronics Trade Handels GmbH
Tel: 99/49/89/577-209
Fax: 99/49/89/570-4379

GAMA

GAMA KOMPJUTERI - STANDARD KOJI PREVAZILAZI VAŠA OČEKIVANJA

Ugled koji GAMA na našem tržištu ima, direktno proističe iz naše neprestane težnje da vam ponudimo kompjuter u koji se bezrezervno možete pouzdati.

Da smo na dobrom putu svedoče:
garancija od 24 meseca
i
naše dugogodišnje prisustvo i stalan rast na jugoslovenskom i nemačkom tržištu kompjutera.

GAMA KOMPJUTERI BRZI SU I SNAŽNI KOLIKO TO VAMA TREBA, A POUZDANI UVEK I PRE SVEGA

ComTec d.o.o.

Novi Sad, Vojvođanskih brigada 7/IX
Telefon/Fax: 021/22.258

u saradnji sa

NYBBLE

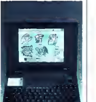
MANUFACTURER OF LAPTOP COMPUTERS AND PERIPHERALS

LAPTOP - NOTEBOOK





NYLAP 286-16/40 MB
NYLAP 386SX -16/40 MB



NB 6300 (286-12/20 MB)
NB 6800 (386SX/20 MB)

NB 8800 (386SX-16/20MB)
samo 2kg!



PREFER 286-12/40
PREFER 386-20/40

Comtec d.o.o. je autorizovani prodavac NOVELL proizvoda
Nudimo Vam i standardne AT 286 /386/486 računare i širok izbor
aplikativnog softvara prilagođenog radu u mreži

Dileri, javite se! Povoljan rabat!

Microsoft

MS DOS 5.0
Windows 3.0
Works

REY
F. O. BEOGRAD

ASHTON·TATE

dBase IV
Applause II
Framework III

NOVELL

NetWare 2.2
NetWare 3.11
ELS II

Fox Software

Fox Base
Fox Pro
Fox Graph

IBM

Hardware

WORDSTAR

Professional 6.0
2000 Release 3.5

BORLAND

Paradox 3.5
Turbo Pascal 6.0
Quattro Pro 3.0

Peter Norton

Backup
Commander
Utilities

Lotus

123 3.1
Simphony
Agenda

PRODUKTI NAJBOLJE KBALITETE KROJENI PO VAŠIM POTREBAMA

REY p.o. Beograd, Glasinačka 31/52, tel. 011/488-5472
Predstavništvo Kragujevac, Bosna Kidrića 35, tel. 034/61-043

SENZACIONALNO!

AT 286 sa VGA kontrolerom i VGA mono monitorom

za $ 820.—



Direktno od proizvođača

80286/12 MH CPU + 1MB DRAM do 4MB
- F.D. 1.2MB ili 1.44MB
- H.D. 40MB
- 2 SERIAL 1 PARALEL FDC IDE ON BORD
- VGA PARADISE 1024×768
- VGA Mono monitor
- Slim case

SOUND BLASTER

NUDIMO ŠIROK IZBOR:
- 386, 486 računara
- Mreže
- Štampače C. ITOH
- Periferije
- Komponente
- Software

SMILEX & ELMIC

POZOVITE NAS! Tel. 011/750-197, 767-516

PHILIPS

predstavlja sa svojim partnerom GAMA

novu generaciju
Philips "BRILLIANCE"
color monitora
bez zračenja
po švedskoj SSI
normi po veoma
povoljnim cjenama.



specijalni dealerski
popusti na sve Philips
monitore i printere

obratite se:

Gama München	Tel. 09 49 / 89 / 57 72 09
	ili 5 70 69 97
	Fax. 99 49 / 89 / 5 70 43 79
Gama Beograd	Tel. 011 / 33 22 75
Misarka 11	Fax. 33 59 02
Gama Zagreb	Tel. 041 / 32 68 41

GAMA

ovlašten distributer

IMTEL

Institut za Mikrotalasnu Tehniku i Elektroniku
Bulevar Lenjina 165 b, 11070 Beograd
Telefon (011)135-420, Fax: (011)138928, Telex: 12797 IMTELYU

5-15% POPUSTA
na računare

RAČUNARE	
PC 386 SX - 16 MHz, 1 MB RAM, HD 40 MB, FD 1 2 MB, Monohromatski monitor, YU tastatura	44.500 din (41.500)
PC AT 286 16 MHz, 1 MB RAM, HD 40 MB, FD 1 2 MB, Monohromatski monitor YU tastatura	32.500 din (29.250)

ŠTAMPAČE	
LaserJet HP IIL, 1MB RAM, Toner	80.000 din.
Epson LQ 1050	40.000 din.
Epson LQ 850	33.000 din.

HARD I FLOPI DISKOVE	
20 MB, 65 ms, MFM interfejs	5.000 din
40 MB, 28 ms, IDE interfejs	10 000 din.
80 MB, 28 ms, MFM ili IDE	18 000 din.
180 MB, 19 ms, SCSI interfejs	32.000 din.
1 44 MB (3 5")	3.000 din.
1 2 MB (5 25")	3.200 din

MONITORI I KARTICE	
VGA 14" monitor i kartica 1024x768	20 000 din.
HERKULES 14" monitor i kartica	5.500 din

OSTALU OPREMU	
Mouse GM - 6 Plus	1 590 din.
Diskete Maxell:	
1 44 MB 3 5", 100 kom.	9 000 din.
360 KB 5.25", 100 kom.	3.200 din.
DOS	2.000 din.
Windows 3 0	4 500 din.
Koprocesor- 80387 25	14.000 din
Modem 2400 MNP5	6 000 din
Eprom programer	4 000 din.
Strimer 60 MB	13 000 din.
Arcnet kartica	2.200 din
Novell Netware 386 ver 3.1	200.000 din.

GARANCIJA ZA RAČUNARE 2 godine !

GARANCIJA ZA OSTALU OPREMU godinu dana

PROJEKTOVANJE I INSTALACIJA MREŽE PO ZAHTEVU !



Ključ vašeg uspeha je možda u našim računarima Proverite !

DA LI ŽELITE DA ZNATE VIŠE I PRE ?

JEDNOSTAVNO: PORUČITE REVIJU!

BYTE
PC MAGAZIN
COMPUTER SHOPPER
DATAMATION
PARADOX USER'S JOURNAL
WORD FOR WINDOWS
PC WORLD
UNIX REVIEW
DESKTOP COMPUTING

područje	različiti naslovi
SOFTWARE	324
PC	595
MIKRORAČUNARI	654
RAČUNARSKA INDUSTRIJA	64
RAČUNARSKA ARHITEKTURA	18
APLIKACIJE	500

i još 130.000 naslove sa drugih područja!
POŠALJITE DISKETU I ZAHTEVAJTE INFORMACIJU!

mladinska knjiga trgovina d.d. Ljubljana, Titova 3

KNJIŽARA
Titova 3, Ljubljana
SUBSCRIPTION SERVICE
Tel: (061) 224-057
P. S:
Još uvek vam možemo obezbediti,
takođe, najveći izbor originalnih računarskih i drugih stručnih knjiga.
Tel: (061) 211-895

Dacom DATA AND TELECOM

11000 Beograd, Trg Republike br. 3, VI sprat
telefon: 011 634-248, fax 011 634-437

IZVOD IZ CENOVNIKA KOMPJUTERA I KOMPJUTERSKE OPREME

		Serija A važi od 01. 06 1991 g	CEHA U DEM		CEHA U DIH	
RB	CODE	KOHFIGURACIJA	vpc	mpc	vpc	mpc
1	A	m/b 286/12MHz/1MB,ATBus HD20,FD5 25, ,MoMo14 ,MoCa,ByCa,DOS	2171 20	2238 34	28225 65	29072 42
2	A1	m/b 286/12MHz/1MB ATBus,HD40 FD5 25, ,MoMo14",MoCa ByCa,DOS	2247 18	2314 96	29213.29	30089 69
3	A2	m/b 286/12MHz/1MB ATBus,HD40,FD5.25,FD3 5,MoMo14",MoCa ByCa,DOS	2423 11	2495 80	31500 45	32445 45
4	A2A	m/b 286/12MHz/1MB,ATBus,HD40,FD5 25 FD3 5 MoMo14 ,mVGA,ByCa,DOS	2633 03	2712.03	34229 45	35256.33
5	A3	m/b 286/12MHz/2MB,ATBus HD40 FD5 25,FD3 5,MoMo14",MoCa ByCa DOS	2629 04	2707 91	34177 47	35202.79
6	A4	m/b 286/12MHz/4MB,ATBus HD40,FD5 25,FD3 5,MoMo14",MoCa,ByCa,DOS	2848 96	2934.42	37036 42	38147 51
7	A5	m/b 286/12MHz/1MB ATBus HD20,FD5 25, ,CoMo14",cVGA ByCa,DOS	2956 92	3045 62	38439 91	39593 10
8	A6	m/b 286/12MHz/1MB,ATBus HD80,FD5 25, ,CoMo14",cVGA ByCa DOS	3032 89	3123.87	39427.54	40610 37
9	A7	m/b 286/12MHz/1MB ATBus,HD40,FD5 25 FD3,5,CoMo14",cVGA ByCa DOS	3208 82	3305.09	41714.70	42966 15
10	A7A	m/b 286/12MHz/1MB ATBus HD40,FD5 25 FD3,5,CoMo14 ,SVGA,ByCa,DOS	3408 75	3511 01	44313 75	45643 18
11	A8	m/b 246/12MHz/2MB ATBus HD40,FD5 25,FD3 5,CoMo14 ,cVGA,ByCa DOS	3416 75'	3519.25	44417 71	45750.24
12	A9	m/b 286/12MHz/4MB ATBus HD40 FD5 25,FD3.5,CoMo14",cVGA,ByCa,DOS	3638 67	3747.83	47302 46	45721 74
13	A10	m/b 286/12MHz/1MB,ATBus HD80,FD5 25, MoMo14" MoCa ByCa DOS	2744.99	2827 34	35684 92	36755 46
14	A11	m/b 286/12MHz/1MB ATBus HD80,FD5.25,FD3.5,MoMo14",MoCa ByCa,DOS	2920 93	3008.56	37972.08	39111.24
15	A11A	m/b 286/12MHz/1MB,ATBus HD40,FD5.25 FD3.5 MoMo14" mVGA ByCa,DOS	3180 65	3224 76	40701 08	41922.11
16	A12	m/b 286/12MHz/2MB,ATBus HD80 FD5.25,FD3.5 MoMo14",MoCa,ByCa DOS	3126.85	3220 66	40649 10	41868.57
17	A13	m/b 286/12MHz/4MB ATBus,HD80 FD5.25,FD3 5 MoMo14 MoCa,ByCa,DOS	3346.77	3447 18	43508 05	44813.29
18	A14	m/b 286/12MHz/1MB ATBus,HD80,FD5.25, ,CoMo14",cVGA ByCa,DOS	3534.70	3640 79	45951 15	47329 69
19	A15	m/b 286/12MHz/1MB ATBus HD80,FD5 25,FD3.5,CoMo14",cVGA ByCa,DOS	3710 64	3821 96	48238 31	49685 46
20	A16	m/b 286/12MHz/2MB ATBus HD80,FD5 25,FD3.5,CoMo14 ,cVGA,ByCa DOS	3918 56	4034 08	50915 33	52442.79
21	A17	m/b 286/12MHz/4MB,ATBus HD80 FD5.25,FD3 5,CoMo14",cVGA,ByCa,DOS	4136 48	4260 58	53774.28	55387.51

NA OVE CENE NEMA DOPLATA

ROBA SE PRODAJE NA RATE

		Serija B	CEHA U DEM		CEHA U DIH	
RB	CODE	KOHFIGURACIJA	vpc	mpc	vpc	mpc
1	B	m/b 286/16MHz/1MB,ATBus HD20,FD5 25, ,MoMo14 ,MoCa,ByCa,DOS	2344 92	2415 27	30463 96	31396 46
2	B1	m/b 286/16MHz/1MB,ATBus,HD40 FD5 25, ,MoMo14",MoCa,ByCa DOS	2424 34	2497 07	31516 39	32461.68
3	B2	m/b 286/16MHz/1MB,ATBus,HD80 FD5.25,FD3.5,MoMo14",MoCa,ByCa DOS	2608 25	2686 50	33907.29	34924.51
4	B2A	m/b 286/16MHz/1MB ATBus HD40 FD5.25,FD3.5,MoMo14 ,mVGA ByCa,DOS	2827.70	2912.53	36760.07	37862.87
5	B3	m/b 286/16MHz/2MB ATBus,HD40 FD5 25,FD3.5,MoMo14",MoCa ByCa,DOS	2825 61	2910 38	35732 80	37834 89
6	B4	m/b 286/16MHz/4MB ATBus,HD40,FD5 25,FD3.5,MoMo14 ,MoCa,ByCa,DOS	3055 50	3147 17	39721 52	40913 17
7	B5	m/b 286/16MHz/1MB,ATBus HD20,FD5 25, ,CoMo14 ,cVGA,ByCa,DOS	3170 46	3265 56	41215 83	42452.31
8	B6	m/b 286/16MHz/1MB ATBus,HD40,FD5 25, ,CoMo14 ,cVGA,ByCa,DOS	3249 87	3347 36	42248 27	43515 71
9	B7	m/b 286/16MHz/1MB ATBus HD40,FD5.25 FD3,5 CoMo14",cVGA,ByCa,DOS	3433 78	3536 80	44639 17	45978 34
10	B7A	m/b 286/18MHz/1MB ATBus,HD40,FD5 25,FD3,5 CoMo14",SVGA ByCa,DOS	3503.83	3608 95	45549.84	46916 34
11	B8	m/b 286/16MHz/2MB,ATBus,HD40 FD5.25,FD3.5 CoMo14",cVGA,ByCa,DDS	3515 91	3621.39	45708 82	47078 02
12	B9	m/b 286/16MHz/4MB ATBus HD40 FD5 25,FD3 5,CoMo14 ,cVGA,ByCa DOS	3737 29	3849 41	48584 75	50042.29
13	B10	m/b 286/16MHz/1MB,ATBus,HD80,FD5 25, MoMo14 ,MoCa,ByCa,DOS	2944.73	3033 08	38281 55	39430 00
14	B11	m/b 286/16MHz/1MB,ATBus HD80 FD5 25,FD3 5 MoMo14' ,MoCa,ByCa,DOS	3012 77	3103.16	39166 08	40341 04
15	B11A	m/b 286/16MHz/1MB,ATBus,HD40,FD5 25,FD3 5 MoMo14 ,mVGA ByCa,DOS	3224 09	3320 81	41913 18	43170.57
16	B12	m/b 286/16MHz/2MB ATBus,HD80 FD5 25,FD3 5 MoMo14",MoCa,ByCa DOS	3222.08	3318 74	41887.02	43143 63
17	B13	m/b 286/16MHz/4MB ATBus HD80,FD5 25,FD3 5,MoMo14",MoCa,ByCa DOS	3443 46	3546 76	44764 95	46107 40
18	B14	m/b 286/16MHz/1MB ATBus,HD40,FD5 25, ;CoMo14",cVGA ByCa DOS	3770 26	3883 37	49013 42	50483 83
19	B15	m/b 286/16MHz/1MB,ATBus,HD80,FD5 25,FD3 5,CoMo14 ,cVGA ByCa,DOS	3807 73	3521 96	49500 46	50985 47
20	B16	m/b 286/16MHz/2MB ATBus,HD80 FD5 25 FD3 5,CoMo14",cVGA ByCa,DOS	4017 03	4137 54	52221 41	53788 06
21	B17	m/b 246/16MHz/4MB,ATBus,HD80 FD5 25,FD3 5,CoMo14",cVGA,ByCa,DOS	4401 43	4533.47	57218.55	58935 11

LAP TOP NOTEBOOK COMPUTERS:

Konfiguracija
M/B HEAT 80C286/16MHz/1MB exp 4MB 80287 socet FD 3.5" HD 40MB 2s/1p I/O, LCD displ VGA 600×400 65 Keys, Weighl 2.9Kg
Options 2400 bps modems card 9600/2400 fax card
CENA mpc 3202 — USD 75857 — DIH

Konfiguracija
M/B NEAT 80C386SX/16MHz/1MB exp 4MB 80387 socet FD 3.5" HD 40MB 2s/1p I/O LCD displ VGA 600×400 65 Keys. Weighl 2.9Kg
Options 2400 bps modems card 9600/2400 fax card
CEHA mpc 3558 — USD 85397 — DIH

ATTACHE CASE SYSTEM
1 — TL 1020/286 Holebook PC computer
2 — TL 100 Car Adapter
3 — TL-200 Portable Printer
4 — TL-405 Pockt Modem and Fax,
5 — TL-411 Mouse
6 — Addifional hahery Packs
7 — Attache Cases
CEHA mpc 4607 — USD 103202 — DIH

Car Adapters
10—15V Vdc input 110V Vac output 60Hz
CEHA mpc 118 — USD 2775 — DIH

Portable Printer
24 dpi thermal type, friction roll feed
6 Vdc NiCd battery with up to 90 minutes
FCC Class B approved, A4 paper size
CEHA mpc 489 — USD 11742 — DIH

Pockel Modem and Fax
Modem speed of 2400 bps CCITT V.21 V.22 V22 Bis Asinhronous data trans Fax trans 4800 bps
CEHA mpc 177 — USD 4227 — DIH

Mouse CEHA mpc 33 — USD 768 — DIH
Add battery Packs CENA mpc 63 — USD 1495 — DIH
Attache cases CENA mpc 223 — USD 5337 — DIH

INSTITUT ZA NUKLEARNE NAUKE "BORIS KIDRIČ", VINČA
CENTAR ZA PERMANENTNO OBRAZOVANJE
11000 Beograd, Nemanjina 4/X
Telefoni: (011) 683-390, 682-486, 641-155/107,181
Telefax: (011)682-486

KNJIGE U OBLASTI PROGRAMIRANJA, PROGRAMSKIH JEZIKA I PRIMENE RAČUNARA

1. **AutoCAD (verzija 10.0)**
konstruisanje i projektovanja pomoću personalnih računara
(šesto izdanje 1991.)
Autori: Boris Damjanović i Petar Damjanović
Latinica 444 strane, format B-5, cena 780 dinara

2. **Uvod u C jezik**
(treće izdanje, 1990.)
Autor: Vladan Vujičić
Latinica, 317 strana, format B-5, cena 585 dinara

3. **Primena programa SYMPHONY na personalnim računarima**
(treće izdanje, 1990.)
Autor: Dragan Pantić
Latinica, 226 strana, format B-5, cena 455 dinara

4. **OS/2 - vodič za korisnike**
(prvo izdanje, 1989.)
Autor: Zorica Jelić
Latinica, 253 strane, format B-5 cena 455 dinara

5. **VENTURA - računarsko izdavaštvo**
(drugo izdanje 1990.)
Autor: Predrag Davidović
Latinica, 253 strane, format B-5 cena. 455 dinara

6. **FORTRAN 77**
standard sa dopunama za personalne računare
(drugo izdanje 1990.)
Autori: Vlajko Kocić i Zoran Konstantinović
Latinica, 422 strane, format B-5, cena. 780 dinara

7. **UNIX - vodič za korisnike**
(drugo izdanje 1990.)
Autor: Zorica Jelić
Latinica 422 strane, format B-5, cena 780 dinara

8. **Primena programa FRAMEWORK III na personalnim računarima**
(prvo izdanje 1990.)
Autor: Dragan Pantić
Latinica, 326 strana, format B-5, cena 585 dinara

9. **PROGRAMSKI ALATI U MATEMATICI**
MathCAD, Grapher, Eureka
(prvo izdanje, 1990.)
Autor: Ante Čurin
Latinica, 402 strane, format B-5, cena. 715 dinara

10. **Primena programa QUATTRO na personalnim računarima**
(prvo izdanje 1990.)
Autor: Dragan Pantić
Latinica, 296 strana, format B-5 cena 585 dinara

11. **DOS ukratko**
(prvo izdanje, 1990.)
Autor: Dragan Pantić
Latinica, 89 strana, format B-5, cena 260 dinara

12. **Vodič za VAX/VMS**
(prvo izdanje 1990.)
Autori: Tamaš Kerepeš, Zvonko Oršolić, Saša Matijević
Latinica 512 strana, format B-5, cena 910 dinara

13. **Primena programa EXCEL na personalnim računarima**
(prvo izdanje, 1991.)
Autor: Dragan Pantić
Latinica, 272 strane format B-5 cena 585 dinara

14. **Unix - vodič za programere**
(prvo izdanje, 1991.)
Autor: Zorica Jelić
Latinica, 326 strana, format B-5, cena. 715 dinara

15. **WINDOWS 3.0**
(prvo izdanje, 1991.)
Autor: Dragan Pantić
Latinica, 273 strane, format B-5, cena. 585 dinara

"RAČUNARI", juni 1991

Poručujem (ispod rednog broja knjige upisati broj poručenih primeraka)

Redni broj knjige	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.
Broj poručenih primeraka															

IME I PREZIME
(Naziv preduzeća) ____________________

ULICA I BROJ ____________________

BROJ POŠTE I MESTO ____________________ Telefon ____________

Porudžbenicu poslati sa petim primerkom uplatnice na adresu:
Institut za nuklearne nauke "Boris Kidrič", Vinča
Centar za permanentno obrazovanje, Beograd, Nemanjina 4/X.
Uplata se vrši na žiro račun:
INSTITUT ZA HEMIJSKU DINAMIKU I PERMANENTNO OBRAZOVANJE, broj: 60803-603-17361.
Troškove poreza na promet i isporuke knjiga snosi poručilac i namiruje ih unapred zajedno sa uplatom za knjige.

Computer hit biblioteka
Poštanski fah 116, 71210 Ilidža
Tel 071/ 270 213 i 071/621-025

Computer hit biblioteka
LITERATURA ZA IBM PC

Novo u izdanju Computer hit biblioteke
Edicija "Džepna knjiga za PC"

1. MS DOS 3 3/4 0

Knjiga neophodna svim korisnicima IBM PC i kompatibilnih računara. Detaljan opis komandi sa svim opcijama koje su na raspolaganju za pojedine verzije MS DOS a. Objašnjenja o primjeni. Posebne napomene za verziju 4 0.

125 strana — 190,00 din

2 PC Tools Deluxe 6 0

Priručnik za rad sa najpopularnijim pomocnim programom za personalne računare. Knjiga sadrži opis instaliranja programa, korisničkog interfejsa, rada sa datotekama i direktorijima kao i objašnjenja za rad sa programima koje sadrži paket PC Tools. PC Shell PC Backup Compress Deskfix, PC Cashe, Desktop

130 strana — 190,00 din

3 WordPerfect 5.1

Detaljan podsjetnik za rad sa tekst procesorom koji je osvojio svijet. Sve opcije programa su date abecednim redom što omogućava njihovo lako nalaženje. Opis rada korak po korak. Nova mogućnost WordPerfecta tabele i formule stilovi i makro naredbe

130 strana — 190,00 din

4 MS Quick Basic 4.5

Referentni priručnik za popularni programski jezik. Opis svih instrukcija i funkcija sa tačnom sintaksom i načinom pozivanja. Ilustrovano mnogobrojnim primjerima.

190 strana — 190 00 din

5 Norton Utilities

Priručnik za rad sa svim komponentama sistema Norton Utilities, od instaliranja programa i testiranja vašeg računara do korištenja naprednih mogućnosti ovog vrlo korisnog programa.

120 strana — 190 00 din

6 Lotus 123 Rel 2 2/3 0

Podsjetnik za rad sa integrisanim paketom programa koji je nametnuo svjetske standarde. Kompletan pregled svih komandi, opcija funkcija i makroa programa

190 strana — 190,00 din

7. Biblioteka Runtime funkcija za MS C

Knjiga koja će vam olakšati rad i ustedjeti trud i vrijeme za razvoj najčešće korištenih funkcija. Jednostavna neophodna svakom C programeru. Detaljan opis više od 300 funkcija

250 strana — 290,00 din

8. MS Windows 3 0

◂ *Windows serija!*

Operativni sistem nove generacije. Budite među prvim koji će ovladati njegovim ogromnim mogućnostima

160 strana — 210,00 din

9 Clipper 5.0

Pregled svih instrukcija komandi i funkcija sa sintaksom namjenom opisom argumenata i primjerom korištenja. Podešavanje okruženja programa instrukcije pretprocesora standardne klase funkcija extend sistema.

320 strana — 330,00 din

10 AutoCad 10

Referentni priručnik za rad sa popularnim programom namijenjenim dizajniranju i projektovanju uz pomoć računara u kom se na sistematičan način nude objašnjenja o radu sa svim komandama programa i raspoloživim opcijama.

280 strana — 330 00 din

11 Venture Publisher 3 0

Referentni priručnik za najpopularniji program iz oblasti stonog izdavaštva (DTP) na PC u. Instaliranje programa sistematizovan pregled svih opcija, opis rada sa tabelama i formulama instalacija fontova. Obuhvaćene su verzije za GEM i MS Windows

250 strana — 330 00 din

12 CorelDRAW! 2 0

◂ *Windows serija!*

Upoznajte se sa fascinantnim karakteristikama najboljeg programa za crtanje i dizajniranje na PC računarima. U novoj verziji donosi rad u 3D perspektivu ovojnica nove fontove mogućnost kreiranja sopstvenih fontova, konverziju fontova u/iz raznih formata, novi trace program i još mnogo puno toga.

190 strana — 190,00 din

U pretplati (izlazi iz štampe 25 06 1991.)

13 QuattroPro 2 0

Referentni priručnik za rad sa jednim od najboljih programa za tabelarna proračune. Dat je pregled komandi funkcija i makroa i opisan rad sa grafikom. Ako već radite sa programom QuattroPro ova knjiga će vam biti nezamjenjiv podsjetnik.

140 strana — 190 00 din

14 Word for Windows 1 1

◂ *Windows serija!*

Upoznajte tekst procesor nove generacije. Iskoristite prednosti grafičke radne okoline WYSIWYG prikaza i štampanja dinamičke razmjene podataka sa ostalim Windows aplikacijama moćnih makroa i ostalih prednosti ovog izvanrednog programa

140 strana — 190,00 din

U pretplati (izlazi iz štampe 15.07.1991.)

15. MS Excel 3 0

◂ *Windows serija!*

Ova knjiga će vas uvesti u sve tajne Excela trenutno najmoćnijeg programa za tabelarne proračune. Jedan pravi izbor za korisnike koji traže snagu ali i fleksibilnost i lakoću upotrebe. Pored osnova rada komandi funkcija i makroa programa, dat je i prikaz pomoćnih alata Solver i Q+E

150 strana — 210 00 din

Knjige možete naručiti poštom – priloženom narudžbenicom ili telefonom na brojeve
071/270-213 i 071/621-025 od 09-17 sati

Computer hit biblioteka Poštanski fah 116 71210 Ilidža

N A R U D Ž B E N I C A

Naručujem sljedeće naslove iz edicije "Džepna knjiga za PC"

(upišite broj komada ispod odgovarajućeg broja knjige)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Prezime i ime ______________________

Adresa ______________________

Mjesto ______________ Telefon ______________ Fax ______________

Troškove poštanskog pakovanja pokrivaju na teret naručioca

AUTRONIC

6/91

Handelsgesellschaft m. b. H.

Computer - Systeme

Radetzkystraße 18 · A 9020 Klagenfurt

☎ (9943) 463 51 48 71
463 51 48 72
Fax 463 51 48 73

Radno vreme
PON - PET
9 - 13 / 14 - 18

Informacije i servis
AUTRONIC Lj, Kardeljeva ploščad 17
☎ (061) 302 581, 302 990, 345 161 (322)
Fax (061) 344-240 Fax & Phone 302 581

Račun kod Karntner Sparkasse /blz 20706
ATS: 900-002534 DEM 9981-484299

Cene su **neto** u **DEM** fco Celovec
Vraćanje Mwst (20%) kod VOLKSBANK
Rosental na Ljubelju (do 20 h)
Zalihe i rokove proverite telefonom.

AUTRONIC COMPUTERS

[illegible]

PEACOCK COMPUTERS ➤ NEW !

[illegible]

LAPTOP COMPUTERS

[illegible]

HARD DISK DRIVES

[illegible]

OPTICAL DISK DRIVES

[illegible]

MONOCHROME MONITORS

[illegible]

COLOR MONITORS

[illegible]

DTP MONITORS

[illegible]

MATRIX PRINTERS

[illegible]

LASER & INKJET PRINTERS

[illegible]

PLOTTERS

[illegible]

VIDEO ADAPTERS

[illegible]

CABINETS

[illegible]

SYSTEM BOARDS

[illegible]

FLOPPY DISK DRIVES

[illegible]

I/O ADAPTERS & MODEMS

[illegible]

KEYBOARDS

[illegible]

RAMS & COPROCESSORS

[illegible]

HD/FD CONTROLLERS

[illegible]

NETWORKS

[illegible]

UTILITIES & OS

[illegible]

SCANNERS

[illegible]

STREAMERS

[illegible]

MICE & DIGITIZERS

[illegible]

ADD-ONS

[illegible]



GENELEC

ELEKTRONIKA & INFORMATIKA

75000 TUZLA
Zlokovac 3B
tel 075/230-878, lokal 126, 231-557
fax 075/237-825, 237-776

11000 BEOGRAD
Jurija Gagarina 115
tel 011/150-949
fax 011/150-949

PROMPTNA ISPORUKA RAČUNARA

PONUDA RAČUNARA

AT 12 MHz/bez DISKA	**21 282 DIN**
slim line 286/12 MHz, 512 KB, FDD 1 2 MB, 2 × ser, + 2 × paralel, c/b monitor, YU znaci	
AT 12 MHz/45 MB	**32 680 DIN**
slim line, 286/12 MHz, 1 MB 45 MB HDD (1 1), FDD1 2 MB, 2 × ser + 2 × paralel c/b monitor YU znaci	
AT 16 MHz/45 MB	**33 901 DIN**
slim line, 286/16 MHz, 1 MB, 45 MB HDD (1 1), FDD 1 2 MB, 2 × ser + 2 × paralel, c/b monitor, YU znaci	
386 SX 16 MHz/45 MB	**44 342 DIN**
slim line 386 SX/16 MHz, 1 MB, 45 MB HDD (1 1), FDD 1 2 MB, 2 × ser + 2 × paralel c/b monitor, YU znaci	
386 20 MHz/85 MB	**62 809 DIN**
Mini Tower, 386/20 MHz, 2 MB, 85 MB 19 ms HDD AT BUS, FDD 1 2 MB, 2 × ser + 2 × paralel c/b monitor, YU znaci	

PONUDA ŠTAMPAČA DATAPRODUCTS

MATRIČNI		
9030	(A, 9 pin, 300 cps, YU, paral)	18 920
9040M	(A, 9 pin, 300 cps, YU, paral)	22 630
9040C	(A, 9 pin, 300 cps, YU paral kolor)	25 252
9034	(A, 24 pin 300 cps, YU paral - ser)	23 940
9044M	(A, 24 pin, 300 cps YU, paral - ser)	28 630
9044C	(A, 24 pin 300 cps, YU paral - ser, kolor)	31 254
8070 Plus	(A, 18 pin 400 cps YU paral - ser)	82 990
LASERSKI		
LZR-650	(8 str/min 512 K)	57 507
LZR-1230 E	(12 str/min, 512 K)	157 065
LZR 1240 E	(12 str/min, 1 MB)	171 310
LZR-1260 E I Postscript	(12 str/min, 4 MB)	301 872
LZR-2656/H P2000	(26 str/min, A, format!)	651 697
LZR-2666 I Postscript	(26 str/min, A, format!)	872 608
LINIJSKI		
LB 325	(300 lpm)	273 823
LB 625	(700 lpm)	308 604
LB 1015	(1000 lpm)	447 564
LB 1515	(1500 lpm)	608 687
LM 300 Table Top	(300 lpm, matrix)	243 224
LM 815	(600 lpm, matrix)	384 762
LM 915	(900 lpm, matrix)	906 049
LM 1200	(1200 lpm, matrix)	1 157 866

U CIJENE NIJE URAČUNAT POREZ NA PROMET

PONUDA DIJELOVA

KUĆIŠTA	DIN
Baby 8 PS 200 W	4 068
Mini Tower 8 PS 200 W	5 746
Midi Tower & PS 200 W	6 238
386 Tower 8 PS 220 W	7 178
486 Tower 8 PS 250 W (sa 2 ventilatora)	14 916
MATIČNE PLOČE	
AT-286/12 MHz	4 068
AT-286/16 MHz	5 289
386 SX/16 MHz	15 730
386/20 MHz	24 409
386/25 MHz	26 102
386/25 MHz Cache 64 KB	32.545
386/33 MHz Cache 64 KB	37.427
486/25 MHz Cache 128 KB	62.379
I/O KARTICE	
AT I/O (2×ser - paralel)	814
Paralel port card	402

KONTROLERI (sa kablovima)	DIN
AT BUS (IDE) Adapter & FDC	922
ESDI WD 1007V SE2 (1 1)	8 950
SCSI WD 7000 FASST	13 832
STRIMERI WANGTEK	
3040 FK 40 MB int. 3 5"	13 533
5099 EK 60 MB int 5 25	24 409
5150 PK 150 MB, int 5,25	39 611
DISKETNI POGONI	
5 25 1 2 MB	2.712
3 5" 1 44 MB	2.858
TVRDI DISKOVI	
ST 157A 1 AT BUS 45 MB/28 ms	9 900
Maxtor 8051A 42 MB/28 ms	10 035
ST 1102A AT BUS 89 MB/19 ms	15 730
Maxtor 7080A 81 MB/19 ms	16 815
ST 1144A AT BUS 125 MB/19 ms	18 958

TVRDI DISKOVI (nastavak)	DIN
Maxtor ND 200A AT BUS 200 MB/15 ms	40.682
Maxtor 4170 ESDI/SCSI 158 MB/16 ms	42 038
Maxtor 4380 ESDI/SCSI 338 MB/16 ms	62.380
Micropolis 1654-7 ESDI 161 MB/16 ms	41.380
Micropolis 1664-7 ESDI 340 MB/14 ms	56 954
Micropolis 1674-7 SCSI 181 MB/16 ms	42.718
Micropolis 1684-7 SCSI 340 MB/14 ms	58.310
TASTATURE	
XT/AT YU/ASCII 102 tastera, klik	1 999
MEMORIJE	
DRAM 4164-10	71
DRAM 41256-10	62
DRAM 41464-10	95
DRAM 44256-10	256
DRAM 411000-07	231
SIMM MODUL 256*9-08	759
SIP MODUL 256*9-08	841
SIMM/SIP MODUL 1 M*9-07	2 658

KOPROCESORI	DIN
80387SX-16	12 747
80387 20	16 218
80387-25	16 500
80387-33	24 382
GRAFIČKE KARTICE	
MGP	930
VGA/16 800 × 600	2 658
Super VGA 1024 768	5 153
MONITORI	
14 Mono amber/cb	4 231
14 VGA Mono 640 480	5 289
14 VGA Mono 1024 768	7.981
14 VGA Kolor 1024 × 768	15 730
OSTALO	
Miš Quick QM E1	1 212
Modem Discovery 2400	4 682

ZASTUPNICI

„C-SOFT"	LESKOVAC,	tel 016/47-067,	fax 016/47-830
„BEE LINE"	NIŠ,	tel 018/323-173,	fax 018/47-424
„SPASCHEX"	NOVI SAD,	tel 021/341-685,	fax 021/341-685
„ARDI ENGINEERING"	PRIŠTINA,	tel 038/22-981	fax 038/23-578
„PIM"	RIJEKA,	tel 051/429-361	fax 051/429-361
„LAKOL"	ŠIBENIK,	tel 059/33-569,	fax 059/28-781

Maxtor Colorado Corporation

Sigurno već znate...

- da su **IDE** hard diskovi potpuno istisnuli "MFM" hard diskove sa tržišta
- da su **IDE** hard diskovi formata 3.5"
- da se **IDE** hard diskovi isporučuju sa fabrički uradjenim low level formatom
- da **IDE** diskovi sadrže kontroler ugradjen na disku sa baferom od 4 do 64 kB
- da se **IDE** hard diskovi prave sa kapacitetom od 40 do 560 MB
- da je američka korporacija MAXTOR lider u proizvodnji **IDE** hard diskova
- da Maxtorovi **IDE** hard diskovi imaju ugradjen bafer (Cache) od 32 kB
- da je Maxtorov **IDE** hard disk 8051A najprodavaniji hard disk

Da li znate...

- da novosadsko preduzeće Global distribuira Maxtorove hard diskove u Jugoslaviji
- da Global na skladištu ima Maxtorove **IDE** hard diskove u velikim količinama
- da Global daje garanciju za hard diskove od 6 meseci
- **da su cene koje nudi Global najniže:**

8051A,	**40MB,**	**28ms**		**7.350,00 dinara**
7040A,	**40MB,**	**17ms**		**7.900,00 dinara**
7080A,	**80MB,**	**17ms**		**11.600,00 dinara**
LXT-213A,	**200MB,**	**15ms**		**24.500,00 dinara**
LXT-340A,	**300MB,**	**13ms**		**42.600,00 dinara**

AT-IDE ADAPTER (2HD/2FD) .... 590,00 dinara
AT-IDE (2HD/2FD) + 2S, 1P, 1G .... 1.090,00 dinara

Diskovi su pakovani: 8051A 9 komada, 7040A i 7080A po 21 komad i LXT-213A i LXT-340A po 2 komada.

GLOBAL **IDE dalje!**

GLOBAL D.O.O.
Novi Sad, Kornelija Stankovića bb.
Tel: 021/323-666 Fax: 021/323-888

AHR PCS

AHR - PCS
Viktringer Ring 41
Klagenfurt, Austrija
tel: 9943-463-513 955
fax: 9943-463-513 790

OSNOVNE PLOČE :	
AT-286/12	166
AT-286/16	206
386SX/18	619
386/25	969
386/25, 32KB CASHE	1366
386/33, 64KB CASHE	1660
TVRDI DISKOVI :	
SEGATE 157A	410
SEGATE 1144A	945
NEC 3741 AT BUS	560
NEC 3661 ESDI	1590
FUJITSU 2611T/S	699
FUJITSU 2613T/S	1590
MONITORI :	
MONOCHROME 14" P/W	178
VGA P/W MONOCHROME	223
VGA MULTISYNC CHEER	843
VGA MULTISYNC TM 1480	1036
DISPLAY CARD :	
MGP CARD	29
VGA 256 Kb 800x600	115
VGA 512 Kb 1024x768	182
VGA 1Mb 1024x768 ET 4000	319
AT IDE BUS CONTROLER	37
AT IDE BUS CONTROLER s 2 ser + 1 par	62
INTERFEJS 2 ser + 1 par	32
FD 1 2 Mb TEAC	135
FD 1 44 Mb TEAC	135
TASTATURA CHICONY-YU	79
TASTATURA CHERY	125
RAM 1 MB	111
GM F - 302 MOUSE	89
PISAČI STAR, FUJITSU, NEC	

Cijene su netto u DEM!
Garancija 12 mjeseci
Servisna mreža u Jugoslaviji!

Kako ćete nas naći: Po glavnoj cesti iz Ljubljane prema centru Klagenfurta. Od podvožnjaka skrenete na trećem semaforu (benzinska pumpa SHELL) desno. Informacije i ponude možete dobiti preko telefona i fax-a. DOBRODOŠLI!!!

PAŽNJA: Tražimo servisere na području Srbije! Mogući interesenti javite se na gornji telefon.

EURUS

Uđite u svijet laserskih pisača po cijeni matričnih

- **HP LaserJet III**
- **HP LaserJet IIP**



Kupcima komplete (laserski pisač, font kazeta i kabel za spajanje pisača i računala) odobravamo značajni popust.

- **EURUS YU font kazete**
  - * veliki izbor latiničnih i ćiriličnih fontova
  - * ušteda vremena i memorije pisača
  - * programska podrška za popularne pakete (MS Word, WordStar, Word Perfect, Ventura, MS Windows 3.0, itd)
- **Obnavljanje toner kazeta**
  - * visoka kvaliteta otiska
  - * broj otisaka jednak novoj kazeti
  - * zamjena oštečenog bubnja
  - * ušteda preko 50 %

Što još nudimo?
Sve ostale HP proizvode, toner kazete, matrične pisače Epson, PC računala, kalkulatore, itd.

NOVOST NA TRŽIŠTU

- **Automat vrećastih otprašivača**
  - * 16 kanala, detekcija kvara ventila (kratki spoj, prekid, mehanički kvar), javljanje greške i alarm, namještanje parametara, zaštita izlaza, itd
- **Detektor vlage**
  - * priključak do 8 senzora, slobodni kontakti prema zvučnom ili svjetlosnom alarmu, itd
- **Uređaj za nadzor šankova u ugostiteljstvu**
  - * kontrolira istočenu količinu pića i cijenu, detektira svako neovlašteno rukovanje, itd
- **Phone-watch**
  - * kontrolira neovlašteno korištenje telefona
  - * bilježi sve brojeve koji su pozivani sa telefona

Posjetite nas ili nazovite.

Autorizirani HEWLETT PACKARD dealer

EURUS d.d
41020 Zagreb
Dubrovečka c. 2
(Avenija B. Kidriče 2)
tel: 041/528 958
fax: 041/524 878

computerworld
Hindenburg str. 56, 6100 Dermstedt
TEL: 9949/6151/31 56 68, 31 56 36
FAX: 9949/6151/31 45 00, 31 34 02

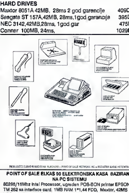
Novo otvoreno predstavnistvo-ured: 41000 Zagreb, Radiceva 11, tel:041/421-973, 435-448
Novo otvorani izlozbani i prodajni prostor; 41000 Zagreb, Margaretska 2/1, tel:041/423-604

POZIVAMO PODUZECA I SERVISERE ZAiNTERESIRANE ZA RAD SA NAMA DA NAM SE JAVE

NASA AKTUELNA PONUDA

MDTHERBDARD	
80286/12Mhz,G-2	175DM
80286/16Mhz, G-2	239DM
80386 /16Mhz,SX	809DM
80386/25 w/o Cache	1089DM
80386/25,64Cache	1409DM
80386/33,64Cache	1559DM
80486/25,128Cache	3095DM
GRAPHIC CARDS	
monochrom+LPT1	34DM
Vge 8bit/256KB	114DM
Vge 16bit/256KB	169DM
Vga 16bit/512KB	199DM
CDNTRDLLER CARDS	
MFM,1,1,2Hdd/2Fdd	89DM
AT-BUS, 2Hdd/2Fdd	34DM
WD 1006MM2	179DM
WD 1006SR2	189DM
WD 1007ESDI	359DM
MULTI I/D CARD	
2ser+1par+1game	25DM
At-bus adap+multi I/D	65DM
XT/Multi I/O card	65DM
FLDPPY DRIVES	
Mitsumi/YE-DATA	
1,2MB ili 1,44MB	119DM
TEAC 1,44MB	134DM
TEAC 1,2MB	129DM
KEYBDARDS	
YU-sa ili bez Click-a	72DM
CHERRY US	129DM
Chicony US	72DM
KUCISTA	
DTP,200W bez LED	135DM
DTP,200W sa LED	199DM
Mini-tower,200W	239DM
TOWER case, 220W	379DM
STREAMER	
40MB Wangtek,int.	595DM
40MB Wangtek,ext	769DM
CDPRDCESDRS	
80267/XL/Intel	389DM
80387/SX/Intel	559DM

HARD DRIVES	
Maxtor 8051A 42MB, 28ms 2 god garencije	409DM
Seagate ST 157A,42MB, 28ms,1god garancija	395DM
NEC 3142,42MB,28ms, 1god gar	475DM
Conner 100MB, 24ms,	1029DM



POINT OF SALE ELKAS 90 ELEKTRONSKA KASA BAZIRANA NA PC SISTEMU

80286/16Mhz Intel Processor, ugraden POS-BON printer EPSON TM 262 sa interface card, 1MB RAM 1*1,44 FDD, Maxtor, 42MB HDD 256KB 8bit VGA, orig IBM monitor 9", 3 slobodne slota za prosirenje, network compatible, ladica za novac, 200W PS — 3798,00DM

LCD DISPLAY 2*16 znakove + interface/OPTREX	195,00 DM
BARCODE LIGHT PEN+DECODER / DATA LOGIC	249,00DM
BARCODE TOUCH SCANNER /DATA LOGIC	1239,00DM
EPSON TM 930 MULTIFUNKCIONSKI PRINTER SA BON-JOURNALOM-SLIP PRINTEROM	1569,00DM
POWER SUPLY ZA EPSON TM 930	259,00DM

Pored toga nudimo rjesenje za mobilnu obradu BAR-COOA, te obradu u okviru svih postojecih mreza.

Maxtor 7080A,82MB, 19ms 2 god gar	695DM
Seagate ST1102, 82MB, 28ms 1god gar	679DM
NEC D5665,179MB, 18ms 1god	1299DM
Maxtor MXT 4380S/E, 382MB,16ms	2649DM
Isporucujemo HDD's CDNNER, Deuntum	

MQNITQRS	
IBM Vga monoc 9"	495DM
Monoc. 14" ili amber	195DM
Monoc VGA 14"	249DM
Mon VGA, MG-11,14"	419DM
Vga Col 800*600, 14"	609DM
Vga Col 1024*768, 14"	649DM
Vga Col 1024*768, 19"	2179DM
TERMINALI	
Wyse 60	1009DM
Qume QVT 101+	869DM
Qume PCT EV	969DM
NETWDRK CARDS	
Ethernet NE1000or	389DM
Ethernet NE2000or	459DM
Ethernet NE1000com	219DM
Ethernet NE2000com	279DM
MEMDRY	
41256/80	3,20DM
511000/80	1,20DM
514256/80	2,90DM
256*9/70 SIMM	34DM
256*9/70 SIPP	36DM
1MB*9/70SIMM	109DM
1MB*9/70SIPP	111DM
MATRIX/LASER PRINTER	
Epson LX400	419DM
Epson LX850	579DM
Epson LD400	619DM
Fujitsu DL1100	869DM
Laser Dume WP+	2089DM
Qume SeriesII,1,5MB	3289DM
HP Laser Jet IIP	2499DM
HP Laser Jet III	3599DM
HP DESKJET 500	1149DM
SCANNERS/MAUSES	
Genius Gmf 302	85DM
Genius Gmf 320	59DM
Logitech 256 Scanner	569DM
Chinon A-4	1029DM
HP Deskjet+	2979DM
Genius GS 4500	289DM
FAX-CARD	389DM

Garantni rok za ponudenu opremu je 12mjeseci. Rok isporuke: sa skladista, za deelera po dogovoru.
Za ponude, eventueine poteskoca obratite nam se direktno ili pak nasim ovlastenim serviserima.

INTERELEKTRONIK STEVANA SUPLJIKCA 45 26000 PANCEVO TEL: 013/511-150, FAX: 521-086	INFORMATICKI INZINJERING, @ POLJAK ZELIMIR GUNDULICEVA 61F 54 000 OSIJEK TEL: 054/41666-276, FAX/TEL: 125-371	ACS gosp. VOJISLAV PAVLOVIC HO-SI-MINOVA 27 SAM KONZALTING CENTAR 11000 BEOGRAD TEL:011/136-322, 135-604 FAX/TEL. 140-027

ŽELITE LI MOĆ

ZA ŠAKU DOLARA?

Upoznajte HAUPPAUGE 4860R matičnu ploču koja PC računalu daje snagu mainframe kompjutera

HAUPPAUGE 4860R udružuje Intel procesore 486 i 860 (RISC).

Rezultat?

16 MIPS i 25 MFLOPS što je 10 puta brže od 486 računala!!! Naravno, sve to uz punu DOS, OS/2, Novell netware 386, Windows 3.0 i UNIX kompatibilnost



IDEALNA OSNOVA ZA

CAD, LAN i DTP,

UNIX radne stranice, sve matematičke i grafičke aplikacije te ekspertne sisteme.

TRAŽE SE DILERI!

Informacije i prodaja

AUTRONIC

YU - 41000 Zagreb, Kollerova 3

Telefon fax 041 232-259

Šta kaže štampa:

O modelu ACS 386/25:

"... Sve to je bilo lepo sastavljeno, izgleda bez žurbe i temeljno. ... radi se o računaru koji je napravljen da traje ..."

računari, br. 68, st. 16-17

O modelu ACS 486/25:

" ... očarava performansama i kvalitetom komponenata, ... Spoj je izvanredan, što ukazuje na to da je neko u ACS-u dobro obavio domaći zadatak ... Ako vam treba jak CAD računar ili pouzdan server, eto mašine koju treba veoma ozbiljno razmotriti."

računari, br. 68, st. 16-17

Ni sami to bolje ne bismo rekli!

ACS Beograd

Advanced Computer Systems

S.A.M. - ACS Beograd
Ho Ši Minova 27, YU-11070 Novi Beograd
Tel. (011) 136-322, 135-804
Fax (011) 140-027

ACS je projekat u razvoju konsalting agencije S.A.M.

KUĆIŠTA SA NAPAJANJEM	DEM	DIN
AT BABY	117	2 379
SUM	154	3 13
MINI TOWER	167	3 39
TOWER	255	5 200
FILE SERVER 375W	950	19 320
WORKSTATION	182	3 522
OSNOVNE PLOČE		
XT 4.77 10 MHz	115	2 342
AT 286 12MHz	145	2 949
N HEADLAND 286-16MHz	185	3 762
N HEADAK 286-20 MHz	235	4 779
386 SX 16 MHz	540	11 011
N 386 SX 16 MHz ALL IN-ONE	599	12 182
N 386 SX 20 MHz	590	11 999
N 386 25MHz CACHE	1 290	27 049
386 33MHz CACHE	1 580	32 347
486 25 MHz	3 400	69 148
486 25 MHz 64 KB CACHE	3 600	73 216
486-25 MHz 64KB CACHE EISA	5 271	107 209
486 33 MHz 128 KB CACHE INTEL	3 900	78 317
DISPLAY KARTICE		
Printer Hercules	28	569
Printer/Hercules/CGA	39	796
VGA 800X600 16 bit	124	2 513
Super VGA 1024x768	160	3 254
GENOA 6200 SUPER VGA	255	5 186
GENOA 6300 SUPER VGA	266	5 404
GENOA 6400 SUPER VGA	420	8 541
GENOA 6400 A SUPER VGA	570	11 592
GENOA 6400 V SUPER VGA	495	10 067
GENOA 6200 VC SUPER VGA	660	13 422
VGA 1280 1024 (NEC 5D)	3 427	69 700
KONTROLERI		
HDD XT MFM	60	1.220
FDD HDD AT MFM 1 1	70	1 423
DTC 7280 AT MFM 1 1	187	3.805
DTC 7287 AT RLL 1 1	197	4.009
AT(IDE) BUS FDD/HDD	35	712
SCSI FDD/HDD	82	1 667
ESDI FDD/HDD	280	5.694
DODATNE KARTICE		
MULTI I/O XT	66	1 333
I/O AT (SER PORT)	22	447
I/O AT (PAR/2×SER PORT)	30	610
I/O AT (PAR/2×SER GAME)	33	671
MULTI USER (4×RS232)	169	3 437
MULTI USER INTELLIG (8×RS232)	713	14 497
AD/DA 12 bits	137	2 780
LAN		
Ethernet compat (NE 1000) 8 Bit	235	4 779
Ethernet compat (NE 2000) 16 bit	280	5 684
Ethernet Pocket Adapter	616	12 522
Ethernet boot rom for NE 1000	10	287
Ethernet boot rom for NE 2000	10	287
IEEE802 3 transceiver	212	8 391
BNC 50 ohm terminator	5	116
BNC 93 ohm terminator	6	118
N-series 50 ohm female terminator	8	174
Cable RG-58 (1M)	3	61
Cable connecter	5	118
Ethernet IEEE802 3 repeater	1 207	24 550
Arcnet coax star LAN card	125	2 556
Arcnet coax bus LAN card	138	2 818
Arcnet twisted pair star LAN card	138	2 818
4 port coax all active hub card	378	8 391
4 port twisted pair hub card	378	3 166
Remote boot rom for arcnet card	10	209
Cable RG 62 (1M)	3	61
TASTATURE		
102 tipke	58	1 176
101 tipka click Cherry YU	76	1 568
101 tipka sa mišem Cherry	187	3 402
101 tipka Cherry	138	2 806
MEKI DISKOVI		
5.25 360 Kb	111	2 257
5.25 1 2 Mb	115	2 338
3.5 1 44 Mb	118	2 338
TVRDI DISKOVI		
Seagate 157A 45Mb/28ms	390	7.931
Seagate — 274A 63Mb 28ms	750	18.514
Seagate — 296N 65Mb/28ms	680	14 033
Seagate — 1102A 83Mb 19ms	710	14 439
Seagate — 1126N 107Mb/15ms	1 406	28 603
Seagate — 1144A 125Mb/19ms	990	20 134
Seagate — 1162A 143Mb/15ms	1 296	26 357
Seagate — 2182E 160Mb/15ms	2 141	43 552
Seagate — 1201A 177Mb/15ms	1 459	30 477
Seagate — 2209N 183Mb/18ms	2 140	43 522
Seagate — 1239A 211Mb/15ms	1 600	32 540
Seagate — 2383A 338Mb/16ms	2 640	53 691
Seagate — 2383E 338Mb/16ms	2.855	58 079
ST 2383N 337 MB 14 MS	2.855	58 079
ST 2502N 442 MB 16 MS	3.569	72.577
ST 4376NV 330 MB 16 MS	3 000	61 013
ST 4385NV 337 MB 10 MS	3 405	69 250
ST 2502NV 442 MB 16 MS	3 735	75 961
ST 4766NV 676 MB 15 MS	3 945	80 232
ST 41200NV 1 050 MB 15 MS	5 713	116 187
NEC		
D 3741 A 44 MB/23 MS	540	10.979
D 3142 44 MB/23 MS	490	9 965
D 3661 135 MB/23 MS	1 350	27 456
D 5855 179 MB/18 MS	1.490	30.303

MLAKAR & CO

AUSTRIJA

Rate ... garanciju a montaža servis u Jugoslaviji. Za savete kod izbora pozovite nas na telefon 9943 4227 2333. Naša trgovina je u Austriji u Podgon (Unterbergen) kraj glavnog puta prema Celovcu 60 km od Ljubljane i 12 km od Ljubelja. Trgovina je otvorena od 8 do 17 časova a u subotu od 8 do 13 časova

FAKS 9943 4227 2091 TELEKS 422749 MLCO A



IZUZETNA PONUDA RAČUNAR + ŠTAMPAČ

PC-M 286-12-45 S ŠTAMPAČEM

– AT 286/12 MHz 1 Mb RAM japanski brzi disk 45 Mb FDD 5,25 1 2 Mb (ili 3 5 1 44 Mb) tastatura 101 14 monohromatski monitor štampač Citizen 9 igl A4

DEM 1.520

DIN 34.400

– s štampačem formata A3

DEM 1.831

DIN 40.400

PC-M 286-16-45-NEAT S ŠTAMPAČEM

– AT 286/16 NEAT MHz 1 Mb RAM japanski brzi disk 45 Mb FDD 5,25" 1 2 Mb (ili 3,5 1 44 Mb) tastatura 101 14 monohromatski monitor štampač Citizen 9 igl A4

DEM 1.590

DIN 35.400

– s štampačem formata A3

DEM 1.901

DIN 41.400

PC-M 386-SX-45

– AT 386 SX/16 MHz -SX 1 Mb RAM, japanski brzi disk 45 Mb FDD 5 25 1,2 Mb (ili 3.5" 1 44 Mb) tastatura 101 14" monohromatski monitor

DEM 1.750

DIN 36 500

Garancija 12 meseci

Mlacom

MLACOM d.o.o
Celovška 185
66100 Ljubljana 1

Tel 061/556-484 Fax 061/556-485

MONITORI	DEM	DIN
... monohromatski	175	3 559
VGA monohromatski	228	4 637
VGA Color 1024×768	596	12 162
... A4 full size VGA	1 540	31 320
NEC Multisync 2A 3D 4D 5D		
VGA CITIZEN 14 1024 × 768	1 200	24 405
PANASONIC 20" 1280 1024	4 427	90 038
MITSUBISHI VGA 720 × 400	975	19 829
MITSUBISHI 14 MULTISCAN	1 350	27 456
MITSUBISHI VGA 14 1024 × 768	3 300	67 114
QUME VGA 14 1024 × 768	1 071	31 320
ŠTAMPAČI		
1 CITIZEN 180D A4	325	6 609
2 C T I 9 Pin A3	636	12 943
Star LC 20	405	8 236
Star LC-15	733	14 907
Star LC-24-200	698	14 195
Star LC-24 15	1 010	20.541
Star ostali modeli		
EPSON FX-1050	945	19 219
EPSON LQ-550	730	14 846
EPSON LQ-1050	1 440	29 286
EPSON ostali modeli		
Laser HP JET II P	2 200	44 743
Laser HP JET III	3 800	77 284
Laser HP JET IIISi		
CANON serija BJ		
QUME serija CRYSTAL PRINT		
CRTAČI		
ROLAND DXY 1100 A3	1 690	34 371
ROLAND DXY 1200 A3	2.088	42 477
ROLAND ostali modeli		pozovite
MODEMI		
2400 int	193	3 918
2400 ext.(MNP5)	243	4 936
9600 ext (MNP5)	1 061	20 047
2400 POCKET	227	4 618
UPS – NEPREKIDNO NAPAJANJE		
UPS 300 VA	480	9 762
UPS 600 VA	570	11 592
UPS 1000 VA	1 097	22.313
RAM		
41256-10	3	61
41256-08	3	61
44256-08	11	223
411000-08	10	203
SIMM/SIP 256K×9-08	35	711
SIMM/SIP 1MB×9-08	97	1.972
COPROCESSOR		
80287	279	5 674
80387SX 16MHZ	650	11 618
80387 25MHZ	790	18 066
80387 33MHz	1 080	21 964
STREAMER		
COLORADO 40/60/120 Mb int	795	16 168
TARGA 150 Mb ext	1 729	35 165
RAZNO		
PC NOTEBOOK XT 20 Mb	1 980	40 472
PC NOTE BOOK 286 VGA 20 Mb	3 700	75 250
FAX NISSEI	1 148	25 567
FAX CARD	557	11 331
FAX MODEM CARD		pozovite
FAX MODEM POCKET		pozovite
Čitač bar koda	466	9 483
Prenosni čitač bar koda	1 042	21 197
CCD Scanner		pozovite
Štampač bar koda		pozovite
Miš Genius 6 Plus	68	1 382
Miš Genius GM F 302	87	1 769
Miš bežični	168	3 428
Track Ball	66	1 333
Tablet Genius GT 1212B 12×12	532	10 819
Tablet Genius GT 1812D	1 027	20 210
Scanner Handy Geniscan GS-4500	290	5 989
Scanner A4 Handy w/pap feeder	1 120	22 779
Scanner EPSON GT-6000 Color	2850	57 962
Eprom UV Eraser	260	5 287
Eprom Writer Card 4x	382	7 989
Disk Box 5×5 25	2	43
Disk Box 10×5 25	4	72
Disk Box 50×5 25	12	244
Disk Box 5×3 5	3	61
Disk Box 10×3 5	3	69
Copy Holder	14	383
Pokrivač za monitor i tastaturu	13	363
Sve vrste EPROM		pozovite

DEM su cene bez poreza na promet. kod Mlakar & CO Austrija
DIN su cene bez poreza na promet kod Mlacom Ljubljana

Na zalihi i druga oprema

DIN su cene bez poreza na promet kod Mlacom Ljubljana

DEM su cene bez poreza na promet kod Mlakar & CO, Austrija

Na zalihi i druga oprema

"PERIHARD"
Poduzeće zadovoljnih korisnika!
Garešnička 36, p.p. 5030, Zagreb
tel 041-263-696, fax:041-263-714

ZAŠTITNI FILTERI ZA MONITORE

Od mikrovlakana » odvode statički elektricitet i niskofrekventno zračenje katodne cijevi
PROTECTOR
V E C
FLEX
» eliminira refleksiju od ekrana

Stakleni filteri
OPTIK GLAS » smanjuje umor i naprezanje očiju
FORTUNA
UP AND DOWN » sve veličine za sve monitore
STAR LIGHT

NAŠ OSTALI ASORTIMAN

Sredstva za čišćenje staklenih filtera
Stalci za printer od pleksiglasa (A3 i A4 format)
FOOT EASE - podložak za noge
ERGO ARM - nosač podlaktice
COPY HOLDER - držač papira uz monitor
Kutije za arhiviranje disketa (5,25" - 100 disketa, 3,5" - 80 disketa)
Kutije za slanje disketa poštom (5,25" - 5 disketa, 3,5" - 6 disketa)

Nazovite nas ili Vama najbližeg redistributera!

ELEKTRODATA Beograd tel (011)340 018
PERIHARD INTERNACIONAL, Novi Sad tel (021) 332 449
INFO D, Beograd, tel (011)609 444, 690 222 (za oba kućni 8146)
CODEX, Titograd, tel/fax (081) 24-149
PC POOL, Sarajevo, tel (071) 219-175, fax.31-760
S.C.T., Čačak tel (032)
OPUS, Kranj, tel (064) 36 785, 324-039, fax 325-879
EKOM, Banja Luka, tel/fax (078)30-781
DEDRA, Osijek tel (054) 120-414 fax 58-877
FENIX, Novo Mesto, tel (068) 26-126, fax 25-145
STRUNA, Ljubljana, tel (061) 320 029

HOUSING ComputerS

Najniže cene - vrhunska kvaliteta !

- **PC kompjuteri**
- **laseri** *Hewlett Packard i Epson*
- **printeri** *Epson i Fujitsu*
- **ink jet printeri** *Hewlett Packard*
- **ploteri i rezači** *Roland*
- **scanneri** *Epson i Hewlett Packard*
- **laptop i notebook kompjuteri**
- **mreže** *Novell i RPTI*
- **hard diskovi** *Fujitsu, Quantum, WD, Maxtor, IBM*
- **grafičke kartice i monitori**
- **koprocesori** *Intel, IIT, Weitek, Cyrix*
- **kartice** *Ethernet*
- **kartice I/O** *(AT, UNIX, IEEE, ...)*
- **izrada programa po narudbi**
- **Svi ostali dodaci za PC !!!**

Nazovite nas još danas, za sve informacije uvijek smo Vam na razpolaganju!

tel/fax: (061) 621 - 145

Sp Pirniče 17/b, 61215 Medvode

MALI OGLASI

Običan mali oglas do 10 reči staje 300 dinara a svaka dalja reč 30 dinara. Centimetar uokvirenog oglasa staje 400 dinara s tim što se mogu zakupiti najmanje 2 cm i što visina ne može biti viši od 5 cm

COMPUTER SERVICE
Vili Vrbik 33a/6 41000 Zagreb
tel 041/539 277 719 892
od 10 do 12 i od 15 do 17 sati
Stranke primamo od 10 do 12 sati!
- SPECTRUM COMMODORE ATARI AMSTRAD
- brzi i kvalitetni popravci
- memorijska proširenja
- prodaja kompjutera flopy drive interface a joystick a
- kablovi za povezivanje sa printerom monitorom televizorom
- ZX centronic interface joystick interface
- C64 eprom moduli centronic kabel
- rezervni dijelovi za kompjutere

Tražite besplatni katalog

NP 46SX! Prodajem kablove za povezivanje sa PC em
Doki Bozo 062/775 183.

DISKETE NAUSHUA BASF
100% error free Made in USA

5 25" 2S/DD (360 Kb)	27 din kom
5 25 2S/HD (1 2 Mb)	40 din kom
3 5 2S/DD (720 Kb)	37 din kom
3 5"2S/HD (1 44 Mb)	65 din kom

Tel (061) 267-632

Prodajem PCXT (512K flopy 360K Herkules monitor) 1000 DM Tel (071/450-428 075/232-638) Zoran (posle 16h)

MS-DOS za početnike
• Knjiga koja se čita u dahu • Štedi vreme energiju novac • Upotreba PC računara posle dan-dva samo 98 din + PT pouzećem • PC računari sa garancijom • Popravka delovi diskete YUSET • Prodaja na tri rate • Tel 015/31-031 016/323 927 022/213-488 078/35 899

HOROSKOPI — najprofesionalniji za IBM-PC Atari Amigu — rasprodajem originale Super brza zarada 037/806 246

NAJZAD KUĆNI PROGRAMI ZA VAŠ PC:

KUFI (Kućne finansije)

KUIM (Kućni imenik adresa)



Naručite **odmah** svoj primerak programa. Po ceni od **480** dinara (porez uračunat) + troškovi poštarine, dobijate disketu sa programom i knjigu - uputstvo Telefonska podrška obezbeđena!

Jugovideo
Prote Mateje 36, 11000 Beograd,
tel/fax: (011) 431-848.

MEŠOVITO PREDUZEĆE
21000 Novi Sad
Todora Jovanovića 14

Tel: 021 396-833
Fax: 021 396-424
Tlx: 64-380

KOMPJUTERI I KOMPJUTERSKI NAMEŠTAJ



ISPORUKA ODMAH

Cene: PC XT od 21.350din, PC AT 286 od 29.950din, 386/SX od 46.150din, 386 od 62.650din, 386/33C od 111.950din, 486 od 179.050din. Povoljne cene LJ printera, plotera, OEM komponenti.
NOVO: PC TELEX KARTICE od 54.300din.
Kompjuterski nameštaj od 2.800 do 6.300din.
Printer stolovi od 5.950 do 6.300din.
Finansijki informacioni sistem 50.000din, Skladišno poslovanje 50.000din, Kamate i krediti 35.000din, Lični dohotci sa kadrovskom evidencijom 40.000din, Posebni finansijski izveštaji 35.000din, Stambena zadruga 35.000din.

ZA NAJAM: 12,7% MESEČNO. SERVIS OBEZBEĐEN

PRODAJEMO PC-AT računare 12 i 16 MHz i štampače EPSON LQ-400 LQ-550 i LQ-1050. OLYMP-ELECTRONIC TEL. 011/400-477

UGRAĐUJEM YU karaktere u sve vrste VIDEO KARTICA i ŠTAMPAČE i zridujem programe za softversko definisanje YU-karaktera za VGA EGA kartice i ŠTAMPAČE TEL. 011 592 087

Prodajem EPROM 27C512 20 Tel 011 157 775 posle 16 00

DISKETE
5,25 in 3,5 2D ili HD sa garancijo prodajem po najugodnijh cjenah. Popust. Tel. (061) 265 525

DISKETE — GARANCIJA

5,25" — 2S/DD (360 Kb)	17 din kom
5,25" — 2S/HD (1,2 Mb)	26 din kom
3,5" — 2S/DD (720 Kb)	23 din kom
3,5" — 2S/HD (1,44 Mb)	39 din kom

Tel (061) 267-632
Na veću količinu popust
Brza isporuka!

ATARI ST/TT
HARDWARE
MEGA 2ST • SM124 MEGAFILE 20/30
PC-SUPERCHARGER, SUPERMOUSE, SCART
ATARI 1040 STE, STFM • SM 124, DISC 3.5"
MEGA STE 2/4/16Mhz, 48MBHD, TT Look!
ATARI TT-030 4/8/8Mb LASER SLM 605
FUJI 3.5" DSDD DISKETE
BORIS GRUDEN, PALMOTIĆEVA 57, ZAGREB
tel. 041/676 228 ili 436-002

INTERSOFT
Andrićev venac 2/1
Računari
PC AT 286/386/486
Telefaxi
Panasonik KX-F90
Laserski štampači
HP LASER JET III
HP LASER JET IIP
Matrični štampači
EPSON (A4,A3)
TEL.
011/331 374
011/340 408
FAX.
011/338 104

Clipper 5.0 | **NiSoft** biblioteka

-BIBLIJA-

Najkompletnije uputstvo za *Clipper* na 620 strana.

Knjigu možete naručiti po ceni od 595 din.:
- pouzećem (telefonom, telefaksom, pismom)
- uplatom na žiro račun (poslati kopiju uplatnice)

Žiro račun "NiSoft" biblioteke
62500-601-18007-80701/39004
Adresa
"NiSoft"
V kongresa KPJ 63/3, 18000 Niš

tel/fax:
018/22-363
telefoni:
018/712-664
018/710-923

!! AKCIJSKA PRODAJA !!

PREDUZEĆE

ADICOM d.o.o.

386 Sx/16

2Mb RAM 44 Mb17 bus, PANASONIC 5.25 1,2Mb MINI TOWER SA DISPLEJEM, 101 USA 14" MONITOR

CENA SAMO 42.990,00

386 SX/16 SUPER CDOR

ISTO + SUPER VGA 1024×768
ISTO + VGACOLOR 14" MONITOR 1024×768

CENA SAMO **58.990,00**

Isporuka odmah
Mogucnost plaćanja u 2 i 3 rate
Tel/fax 061/444-569 od 9—17h
Tel: 018/336-293 od 18-20h

NAMA NE TREBA VEĆA REKLAMA!

Nudimo BATERIJE Nikel-Kadmijum (Ni-Cd) i Litijum (LiSOC12) za razne upotrebe i sa raznim izvodima. P.P. „Computer Electronic", Ul. Novobrdska 22, 38250 Gnjilane. Tel. 0280-28-106

YU latinica, ćirilica, hardverska ugradnja. Matrični štampači, video kartice. 011/347-508; 403-228

Turbo Pascal - naredbe i objekti

Autor Duško Savic
Jednostavan uvod u objektno programiranje!

Nasleđivanje, virtuelni metodi, klase, pisanje novih objekata, opis svih 300 naredbi, prevod svih grešaka, ilustracija DOS i BIOS poziva, opis direktiva prevodiocu, testiranje programa. Svi primeri napisani su korišćenjem objekata.

Priručnik za verzije 5.5 i 6.0 - preporučuje ga i Borlandov zastupnik za Jugoslaviju!

Strana 240, januar 1991. ISBN 86-901 167-1-0
Cena 300 dinara + poštarina, isporuka pouzećem - istog dana!

PC Program, Generala Hanrisa 20,
11040 Beograd, tel/fax (011) 463 296

NOVO NA TRŽIŠTU!!

Za samo 3500 dinara dobijate aparat MAXIPRINT koji obnavlja Vaše trake. Ne zaboravite da sa 10 obnavljanja MAXIPRINT postaje besplatan. Ne bacajte korišćene trake.

Kupite MAXIPRINT i uštedite 95% od troškova koje plaćate za trake.

Radi sa svim trakama za štampača. Ako ne verujete u njegovu efikasnost pošaljite jednu traku i mi ćemo je besplatno osvežiti. Javite se na
MIKROSAM 97500 Prilep Đore Đoreski 5 ili na telefon 098/34 072 098/29 965 098/30-007



MAXIPRINT dobijate poštom „pouzećem"

ELEKTROTEHNA · ELzas

Savska 28/3 (Cibona centar)
41000 Zagreb
tel. 336-070, 336-071, 336 077
fax. 336-072

Poljanska cesta 28
61000 Ljubljana
tel. 061/318 681
fax. 061/328 744

CalComp

PEN PLOTERI A3-A0 A0 s rolom
ELEKTEROSTATSKI PLOTERI crno bijeli i u boji 400x400 dpi 1024 boje
DIRECT IMAGE PLOTERI A1-A0 rad u dvije boje rezolucija 406x406 dpi rola 61 m
HARD COPY (direktno preslikavanje slike u boji sa ekrana) A4-A3 formata
DIGITALIZATORI svih formata (visoka preciznost)
GRAFIČKE KARTICE visoke rezolucije 1280x1024 s grafičkim procesorom
COLOR MONITORI visoke rezolucije 20"
SCANNERI do A0 formata
PC - računala
CADVANCE-CAD software
Potpuna podrška u arhitekturi i građevinarstvu obuka za korisnike kompjutera i projektante

Posjetite nas u našem demo centru

RAČUNARI SU VAŠI!
JAVNO IZVLAČENJE DOBITNIKA NAGRADA
„JUGOEXPORT", BEOGRAD, KOLARČEVA 1 — 28. JUNA U 17.00

KADA SE NA TRŽIŠTU POJAVE JEFTINIJI RAČUNARI PRODAVAT ĆE IH profesional

PC AT 286/12/1-45
Main board 286/12 EMS, RAM 1Mb, MGP YU set 14" Mono monitor, HD/FD Controler, Hard disk 45Mb 28ms, Floppy 1.2Mb, Baby Case + 200W PS, 102 Keyboard YU
29.900 din

PC AT 386/25/2-85
Main board 386/25, RAM 2Mb, MGP set 14" Mono monitor, HD/FD Controller, Hard disk 90Mb 18ms, Floppy 1.2Mb, Mini Tower Case + 200W PS, 101 Keyboard YU
59.981 din

PC AT 286/16/2-40M GRAFIČKA STANICA
Main board 286/16 EMS, RAM 2Mb, Card VGA (1024x768) 14" VGA MONO Monitor 1024x768, HD/FD Controler, 2s/1p port, Hard disk 42Mb 17ms, Floppy 1,2Mb, Slim Line Case + 200W PS, 101 Keyboard YU, Genius Mouse GM6 Plus
39.755 din

...A ZA ONE KOJI NISU TOLIKO BOGATI...

PC AT 388/33/4-125S GRAFIČKA STANICA
Main board 386/33, 128K Cache, RAM 4Mb, Card VGA (1024x768), 14" VGA COLOR Monitor 1024x768, HD/FD Controller 2s/1p port, Hard disk 125Mb, 19ms, Floppy 1.2Mb, Mini Tower Case + 200W PS, 101 Keyboard USA/YU Cherry, Genius Mouse GM6 Plus
91.683 din

PC AT 486/33/8-338C FILE SERVER
MB EISA 486/33 128K Cache, Coproc, RAM 8Mb, MGP 14" Mono Monitor, ESDI HD/FD Controller Adaptec 2322 D, Hard disk CDC 338Mb 14.5ms, Floppy 1.2Mb, Tower Case + 220W PS, 101 Keyboard Streamer 250Mb
198.532 din

ISPORUKA DO 14 DANA

GARANCIJA 18 MESECI

LAPTOP, NOTEBOOK
PLOTERI, SUMMAGRAPHICS, ROLAND, CALCOMP
STAMPACI: Matrični, Bubble Jet, Laserski
EPSON, CANON, FUJITSU, HP
NOVELL mreže 286 V 2.2 5-100 korisnika
NOVELL mreže 286 V 3.11 20-250 korisnika
UNIX, PC MOS Multiuser
INSTALACIJA Ethernet, Arcnet, Multiuser

profesional Ljubljana d.o.o.

TELEFON/FAX. (061) 556-480
TELEFON 559-373 lok. 347, 350 361
STEGNE 19, LJUBLJANA

PROGRAMERI!
OPERATERI!
HACKERI!

Prvi put u Jugoslaviji masaže za one koji sede za računarom (noćima, danima, nedeljama ...) i tako sedeći dobijaju

SKOLIOZU
KIFOZU
IŠIJAS
LUMBOIŠIJALGIJU
GLAVOBOLJE
DISKOPATIJU
DISKUSHERNIJU
OSLABLJENU CIRKULACIJU
BOLOVE U NOGAMA
BOLOVE U VRATNIM MIŠIĆIMA

SVE TO OTKLANJAMO (I SPREČAVAMO) SA GARANTOVANIM USPEHOM – CENTAR PET!

EKSKLUZIVNO NUDIMO:

TRETMANE ZA POJAČAVANJE KONCENTRACIJE

RELAKSACIONE MEDITACIJE PODREĐENE BAVLJENJU RAČUNARIMA!

SVE VRSTE TRETMANA KOJI OTKLANJAJU COMPUTER STRES

Prva jugoslovenska agencija za tretmane koji objedinjuju primenu manuelnih, koncentracijskih i relaksacijskih tehnika:

*HIROPRAKTIKA
*MASAŽE
*PSIHOENERGOTERAPIJA
*MUZIKOTERAPIJA
*RADIESTEZIJA

Cpet

Centar PET
Hotel PUTNIK
Palmira Toljatija 9
11070 Novi Beograd

telefon:011/690-085

AGOSTOL GOAP

servis

Naši ovlašćeni servisi su u:
Subotici, Vinkovcima, Slavonskom Brodu, Sisku, Bjelovaru, Bihaću, Varaždinu, Zaboku, Zagrebu, Karlovcu, Splitu, Zadru, Rijeci, Bujama, Slovenskoj Bistrici, Šmarje-Sapu, Ljubljani, Ajdovščini i Novoj Gorici

Z NAŠEG PROGRAMA NUDIMO VAM PO POGODNIM CENAMA KOMPLETNE RAČUNARE:

- GOAP PC – AT 286-12/H20-65M 34.300
- GOAP PC – AT 286-12W/H40-28M 38.300
- GOAP PC – AT 286-16W/H40-28M 39.800
- GOAP PC – AT 386-16S/H40-28M 52.000
- GOAP PC – AT 386-25/H40-19M 77.500
- GOAP PC – AT 386-25C64/H40-19M 83.400
- GOAP PC – AT 386-33/C64/H40-19M 103.900
- GOAP PC – AT 486-25/H40-19M 189.700

Z plaćanje u devizama dajemo dodatni popust ... kupis



...postoje problemi koje može da reši samo

GOAP

GOAP-GOSTOL, AUTOMATIZACIJA PROCESA, PRVOMAJSKA 124, 65000 NOVA GORICA, TEL.: (065) 26-511, 26-522, 23-411, FAKS: (065) 26-566

COMPUTER & COMMUNICATION GmbH

Grevenbroicher Str. 34, D-4050 Mönchengladbach W GERMANY
Tel: 02161 / 48 17 Fax 02161 / 46 17 68





SISTEM CC-286/12F1MR DM 1.598,-
286/12 MHZ, LM 5,9 1MB RAM SIP MOP AT FDD/HDD KONTROLER (1:1) FD 1,2 MB TASTATURA BABY AT KUTIJA SA DIGITALNIM BIS PLEJOM MONITOR 4 HARD DISK 45 MB 28 MS

SISTEM CC-386SX/20F1MR DM 2.298,-
386SX/20 MHZ LM 26,0 OSTALO ISTO KAO GORE

SISTEM CC-386 33CF4MR DM 3.498,-
386/33 MHZ 64K CACHE LM 68,1 4MB RAM TOWER KUTIJA OSTALO ISTO KAO GORE

SISTEN CC-486/25CF4MR DM 5.549
486/25 MHZ, 128K CACHE LM 117,0, 4MB RAM TOWER KUTIJA OSTALO ISTO KAO GORE

HARD DISK 44,8 MB 28 ms	DM	498,-
HARD DISK 85 MB 28 ms	DM	888,-

MOTHER BOARDS

MB-AT386/33C CACHE 64 KB	DM	1.599,-
MB-AT486/25C CACHE 128 KB	DM	3.988

JEDINSTVENA PRILIKA
SISTEM CC-286 12F1MR + 14"MONOHROMATSKI MONITOR + HARD DISK 44,8 MB 28 ms
DEM 1.599,-

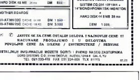
JAVITE SE ZA CENE OSTALIH DELOVA I NAJNOVIJE CENE !!!
RACUNARE PRODAJEMO I U DELOVIMA
POVOLJNE CENE ZA DILERE / DISTRIBUTERE / SERVISE

DETALJNIJE INFORMACIJE MOZETE DOBITI I PREKO NASEG ZASTUPNIKA
GORD SYSTEMS C.O. 91000 SKOPJE RUZVELTOVA 8 GAL 8, YU
TEL 091/233-418 FAX 091 234-808 TLX 51555

91.01

MIKROSAM

DATA PROCESSING AND VIDEO DISTRIBUTION GJORE GJORESKI 5 97500 PRILEP
PHONE FAX (9198) 14 072 30 097



TRGOVCI, UGOSTITELJI, SUPER MARKETI, ROBNE KUĆE, VELIKI MAGACINI!!!

IDEALNO REŠENJE ZA SVE VAS. NUDIMO REGISTAR KASU KOJA JE TZV. POS (POINT OF SALE) TERMINAL. TAČKA PRODAJE VODI TRGOVAČKU KNJIGU, OBRAČUNAVA POREZE, VODI ZALIHE, IZDAJE PARAGON BLOK. ZAŠTITITE SEBE I VAŠE MUŠTERIJE. NUDIMO SLEDEĆU STANDARDNU KONFIGURACIJU

IBM AT COMP. PC 80286 1 Mb RAM, 1.44 Mb FLOPPY
CRT 9" MONOCHROME MONITOR
POS KEYBOARD 88 KEYS (PROGRAMMABLE)
DOT MATRIX MERCHANISMS HIGH SPEED PRINTER
-CUSTOMER ROTARY DISPLAY
-40 MB HARD DISK DRIVE
2X RS 232 C SER. PORT 1 X PAR. PORT

PORED STANDARDNE KONFIGURACIJE NUDIMO I NIZ DODATNIH UREĐAJA. KASU ISPORUČUJEMO SA PROGRAMOM ZA VOĐENJE TRGOVAČKE KNJIGE.

JAVITE SE!!!. MI REŠAVAMO VAŠE PROBLEME!!!

PERI HARD YUGOSLAVIA

"PERIHARD"
Poduzeće zadovoljnih korisnika!
Garešnička 36, p.p. 5030, Zagreb
tel 041-263-696, fax 041-263-714

NOVOSTI U "PERIHARD-u"



* HP LaserJet IIP
* HP LaserJet III
* HP LaserJet IIID
* HP LaserJet IIISi
* HP ScanerJet Plus
* HP Tehnički i poslovni kalkulatori

Osim toga, nudimo Vam i sav potreban pribor

HP RAM proširenja
HP toner kazete
Kablovi (Cent., RS232)
HP PostScript kazeta

Posebno napominjemo
PUNJENJE PRAZNIH TONER CARTRIDGEA
Crno punjenje
Punjenje **u boji** (smeđi, plavi... toner)

VELIKI IZBOR YU-FONTOVA
ZA LASERSKE PISAČE

FONT-CARTRIDGE
(printerski fontovi)

SOFT-FONTOVI
(ekranski i printerski fontovi)

МИ ИМАМО И ЋИРИЛИЧНЕ ФОНТОВЕ!

PERI HARD YUGOSLAVIA

OVLAŠTENI PRODAVAČ
HEWLETT PACKARD
AUTORIZIRANOG DILERA

ELHARD

V. VLAHOVIĆA 46 25230 KULA
Tel. 025-722-740
Fax. 025-723-433

AT 286-12MHz (16MHz)

- Kućište AT Baby
- Osnovna ploča 286-12MHz OWS
- Ram 1MB
- Kontroler FDD/HDD 1:1 IDE
- Flopi MITSUBISHI 1.2MB ili 1.44MB
- I/O 1P/2S/1G
- Grafička kartica Hercules
- Tastatura 101
- Monitor COMPAL Dual Frequency 14"
- Hard disk Western Digital 43MB

CENA: 29.175,00

XT 8088-12MHz

- Kućište AT Baby
- Osnovna ploča 8088 12MHz sa 640KB
- I/O 1P/2S 1G (baterija, sat)
- Kontroler FDD
- Flopi MITSUBISHI 360KB
- Grafička kartica HERCULES
- Tastatura 101
- Monitor COMPAL Dual Frequency 14"

CENA: 15.411,00

- Kutija za 100 Disketa 403,00
- Disketa MAXELL 360KB 38,00
- Disketa MAXELL 1.2MB 68,00
- Disketa SONY 360KB 68,00
- Disketa SONY 1.2MB 68,00
- Neprekidno napajanje 500 W 14.560,00
- Papir za Fax 210×50M 180,00
- Štampač STAR A4, 180cps, 9pin 8.781,00
- Ribon FUJITSU DX 2200 258,00
- Monitor SVGA 1024×768 0.28 .... 15.680,00

DODATNA OPREMA ZA RAČUNARE; SKENER, MIŠ, RIBONI. . .
ISPORUKA ODMAH.
RAČUNARE PRODAJEMO I U DELOVIMA.
DILERI — JAVITE SE!
GARANCIJA 12 MESECI.
VRŠIMO SERVISIRANJE SVIH VRSTA RAČUNARA

· VANGUARD ·
CONSULTANTS LTD misli na VAS !!

MANJE POSREDNIKA-NIŽE CENE

AT286-12 1399DEM

AT286-12+1MB	1399 DEM
AT286-16+1MB	1450 DEM
AT386SX-16+1MB	2010 DEM
AT386-25+2MB	2560 DEM
AT386-25+2MB+64K cache	2960 DEM
AT386-33+4MB+64K cache	3290 DEM
AT486-25+4MB+128K cache	5220 DEM

standardna konfiguracija AT baby+200W, 14"monitor, hercules+printer port, 1.2MB FDD, 44MB HDD, AT-bus 2*FDD/2*HDD, 101 ASCII-klik

doplata za VGA COLOR 640 DEM

Sklapanje i testiranje računara, URAČUNATO U CENU !!

Osim toga širok izbor monitora, štampača i ostalih proširenja.
Roba se može podići u bescarinskoj zoni Segedin (Mađarska),
SAMO 15Km OD JUGOSLOVENSKE GRANICE !

EPSON FX1050 990DEM

Garancija se ostvaruje u JUGOSLAVIJI (12 meseci)

OD SADA MOGUĆNOST KUPOVINE I U JUGOSLAVIJI ZA DINARE
Sve ostale informacije u vezi sa kupovinom računara možete dobiti u Jugoslaviji na telefonu **024/28-246** od 18 do 21 časova

NOVE KNJIGE U IZDANJU Edicije KOMPJUTER BIBLIOTEKA

1 TURBO PASCAL 5.5 (450 str.) 300 din
Posle VI izdanja knjige Turbo Pascal 3.0 izašla je prva knjiga za Turbo Pascal 5.5 vodič za korisnike. Program koji je postao standard i knjiga koja vam standard približava

2 AMIGA GRAFIKA (250 str.) 250 din
Za one koji idu dalje. Odgovori na pitanja kako se kreira korisnički ekran, kako se crta i manipuliše blokovima slika. Poseban dodatak: Grafički programi za Amigu

3 C korak po korak (310 str.) 250 din
Od generalne forme C programa nizova funkcija pokazivača struktura itd. do dva dodatka: Aztec C v 5.0 i sve funkcije Turbo C v.2.0. Knjiga je namenjena vlasnicima PC Amiga i Atarija. Preporučujemo vam svoja ponovljena izdanja

4	Amiga priručnik sa BASIC programiranjem IV izmenjeno i dopunjeno izdanje	250
5	AmigaDOS principi i programiranje IV izdanje. Zajedno sa priručnikom predstavlja celinu	250
6	Amiga/Atari/PC Modula-2 (II izdanje)	250
7	Atari ST priručnik i korak dalje (II izdanje) Knjiga potrebna svakom vlasniku Atari ST računara. Obilje slika, paralelno objašnjene poruke na engleskom i nemačkom jeziku	220
8	ATARI STA GFA BASIC Korak po korak (II izdanje) Na 308 stranica detaljno objašnjen GFA BASIC Detaljno objašnjene komande uz obilje primera	220
9	ATARI ST GFA BASIC Programerski vodič (II izdanje) Obilje programa, procedure, saveta i rešenja	220
10	Atari ST pogled unutra NOVA KNJIGA. Trenutno najtraženija. Interna konfiguracija, memorijske strukture.	220
11	MS—DOS 3.3 (320 str.) IV izmenjeno i dopunjeno izdanje. Standardni kompletan priručnik.	250
12	Amiga/Atari/PC Word Perfect II izdanje Kompletno uputstvo za korišćenje programa	220
13	Commodore 128 programerski vodič	200
14	CP/M sistemsko uputstvo 2.2 i 3.0	200
15	CP/M softver u praksi	200

BITNO: KOD KUPOVINE KOMPLETA OSTVARUJETE VELIKI POPUST

a)	Programski jezici (TURBO PASCAL, C, MODULA II)	650
b)	AMIGA (Priručnik, DOS, Grafika)	580
c)	ATARI (Priručnik, GFA BASIC, GFA ProG.V, Pogled un.)	700

NOVO: BORLANDOVI PAKETI: TURBO PASCAL 6.0 (2 900), C++ 2.0 (8 590), QUATTRO PRO 3.0 (6 990), PARADOX 3.5 (7 990)

Da ne biste oštetili časopis, porudžbinu šaljite na dopisnici, razglednici ili u pismu na adresu: Edicija KOMPJUTER BIBLIOTEKA, F. Filipovića 41, 32000 Čačak
Telefoni (032) 32-322 ili 23-120

INFOTRADE

Mikroračunarske mreze

NOVELL

Školovanje

INFOTRADE Izobraževalni center u Kopru, Vojkovo nabrežje 30 a organizira sledeče tečaje za NOVELL-ove mikroracunarske mreze u mesecu JUNU.

TEČAJ	*DNI*	*VI*
Novell mikroračunarske mreze	*2*	*04 6*
286 — Upravljaci mikroračunarskih mreza	*3*	*11 6*
386 — Upravljaci mikroračunarskih mreža	*3*	*25 6*
Instalacija in tehnična podrška	*2*	*18 6*

Vase cenjene prijave i sve informacije o tečajevima možete dobiti na adresu

INFOTRADE Koper
PE KRANJ
Jaka Platiše 13
64000 KRANJ

TELEFON: 064/329-523
TELEFAX: 064/323 582

KNJIGOVODSTVENI PROGRAMI ZA PC RAČUNALA ZA KRAJNJE KORISNIKE I DISTRIBUTERE

- GLAVNA KNJIGA S KNJIGOVODSTVOM TROŠKOVA
- SALDA-KONTI KUPACA/DOBAVLJAČA S IZVODOM OTVORENIH STAVAKA
- FAKTURIRANJE
- ROBNO KNJIGOVODSTVO
- SKLADIŠTE MATERIJALA
- OSOBNI DOHOĆI
- OBRAČUN KAMATA
- OSNOVNA SREDSTVA
- SITNI INVENTAR

— jedno— i višekorisnička verzija
— puni Source Code
— neograničeno pravo na distribuciju exe verzija programa
— mogućnosti a ne ograničenja
— moderan design
— jednostavna obuka i upotreba
— modularnost
— fleksibilnost
— svi izlistaji i na ekranu i na štampaču
— svi standardni štampači
+ mogućnost specijalnih štampača
— detaljna uputstva za upotrebu (na disketama i štampano)
— detaljna tehnička dokumentacija
— tehnička podrška

Detaljne informacije na tel. 041/535-920
041/535-922
fax 041/535-920
041/535-922

G&G® electronic

Savska cesta 41
pp 45
41000 ZAGREB

HARDLOCK E-Y-E·
NAJEFIKASNIJA ZAŠTITA PROGRAMA OD ILEGALNOG KOPIRANJA

NOVELL — VODEĆI SVETSKI PROIZVOĐAČ MREŽNIH PROGRAMA

Authorised Dealer

i ovlašćeni prodavac mikronova vam nude:
kompletnu paletu NOVELOVIH proizvoda, instalaciju i održavanje

DODATNI PROGRAMI ZA MREŽE

- udaljene radne stanice (uključivanje u mrežu posredstvom telefonskih linija)
- uključivanje u paketske komunikacione mreže (YUPAK)
- povezivanje mreže sa velikim sistemima
- povezivanje više mreža

KOMPLETNA PONUDA RAČUNARSKE OPREME

- IBM kompatibilni računari: AT, 386, 486
- štampači: matrični, heavy duty i laserski
- mrežne kartice i ostala mrežna oprema, modemi
- rezervni delovi
- data switch
- ...ostali proizvodi vrhunske tehnologije

POTROŠNI MATERIJAL

- trake za štampače
- perforirani papir
- streamer trake
- diskete
- toneri

mikronova

PROJEKTOVANJE I IZGRADNJA INFORMACIONIH SISTEMA

APLIKATIVNI PROGRAMI I PROGRAMSKI PAKETI

● materijalno i finansijsko poslovanje ● kadrovska evidencija, lični dohotci i krediti ● uprave društvenih prihoda ● devizno poslovanje ● poliklinička i bolnička evidencija ● advokatske kancelarije ● obrada čekova ● obračun kamata ● budžetsko knjigovodstvo ● statistika ● trgovinska preduzeća ● autotransportna preduzeća ●stambena preduzeća ● uvozne/izvozne carinske deklaracije ● plan i analiza

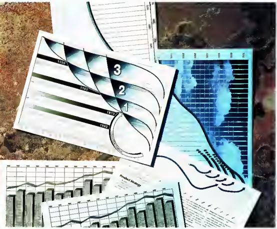
PROVERITE NAŠU REFERENTNU LISTU

DILERI DOBRODOŠLI!

POZOVITE NAS
P.P. mikronova D.O.O. Novi Sad
Vojvođanskih brigada 7/IX
tel: 021/21-821 ili 611-671, fax: 021/28-042

POSETITE NAŠU NOVOOTVORENU PRODAVNICU
ul. Partizanskih baza 21, Novi Sad

Najbrži u LaserJet porodici



Zahvaljujući korištenju suvremene tehnologije zasnovane na snažnom RISC-procesoru, najnoviji model Hewlett-Packard laserskog pisača, HP LaserJet IIISi, postiže brzine ispisa od dosad nevjerojatnih 16 stranica u minuti. Za zadovoljenje ovakvih eksploatacijskih mogućnosti, korisnicima su na raspolaganju dvije kasete kapaciteta od po 500 listova, posredstvom kojih se bez problema ostvaruje i dvostrani tisak. Pri tom svako otisnuto slovo odlikuje izuzetna oštrina i glatkoća linija, koju uz novoprimjenjeni toner ekstrafine granulacije dodatno poboljšava i primijenjena tehnologija poboljšane razlučivosti (RET Resolution Enhancement Technology). HP LaserJet IIISi također omogućuje izravno povezivanje s računalnim mrežama poput Novell i sl.

Kao i prethodni modeli LaserJet serije, i ovaj pisač podržava rad s grafičkim



jezikom PCL5. Pritom u radu koristi standardno ugrađena pisma fleksibilnih veličina, te vektorski grafički jezik HP-GL/2.

Osim toga postoji i mogućnost ugradnje opcije za rad u skladu s Adobe PostScript grafičkim standardom uz odgovarajuću kompatibilnost oba grafička jezika.

Zahvaljujući navedenim mogućnostima HP LaserJet IIISi danas nedvojbeno daleko nadmašuje slične uređaje na tržištu, garantirajući i nadalje apsolutni primat Hewlett-Packard laserskih pisača.

• Za dodatne informacije pozovite Hermes plus 061/552-941

ZVANIČNI DEALERI: IPP Sistemi 011/764 802, CORES 011/661-053, INFOTRADE 038/25 830, EXTREME 061/301-530, TREND 063/851-610, MAC ADA 061/329-877, ATR, 061/327-068, HERMES 061/321-445, MDS, 061/328-475, ELLMAX 062/813-975, STING 061/446-033, KERN Sistemi, 061/224-543, EURUS 041/528-958, DATA COMMERCE, 041/333-533, MICROLAB 041/692-704, DEDRA, 054/120-414, ELCOTECH, 071/274 912, IDENT, 078/32-671, COMEL, 071/656-120, SANOSOFT, 091/263-051, INPROCOM, 0671/54-100

MOGUĆNOSTI SU SE OSTVARILE


ТО СМО ЧЕКАЛИ ГОДИНАМА!
Windows 3.0 (YU fonts)
MS Word 5.50
Quick C, Quick Basic
Flight Simulator
Quattro Pro
BORLAND
Microsoft
Litron
HEWLETT PACKARD
Laser Jet IIP
Кад ти треба – свратиш до радње па купиш. Платиш у динарима, наравно!
Нема планирања, распитивања, путовања, трагања, транспортовања, царине...
Једноставно – свратиш и изабереш. А, ако си у недоумици,
имаш кога да питаш за савет. Ту је и гаранција и сервис
и све остало.
Ни цене нису ништа горе него тамо.
То је то!
Радња Радно време од осам до осам.
Рачунари, компоненте, програми,
прибор, папир, литература...
LX, FX, LQ
EPSON
IBM
PS/1
COMPYUTER line
Плус 62 ћирилична фонта!
Ново!
процесор 286
1 MB RAM
30 MB хард диск,
3.5" флопи диск,
VGAграфичка
монохроматски
монитор
(опција колор),
тастатура, миш,
MS DOS 4.01,
Microsoft WORKS
(фајл менаџер,
текст процесор,
спредшит и база
података)
Прва из низа специјализованих
југословенских продавница рачунара
отворена је у Београду,
у улици Кнеза Милоша 4
(одмах изнад Лондона).
UNIVERZAL
jugodata
ОПЕТ НЕШТО НОВО!
11000 Београд, Саве Ковачевића 6, Телефон: 444-31-76, Телефон/Факс: 453-560

D M P™-6 0   D L   S E R I E S



Jednostavno rečeno, šijemo konkurenciju

Da rešimo ko je najbolji.

BRZINA. U nedavnom ispitivanju Houston Instrument DMP-61 DL se pokazao kao najbolji od tri najprodavanija plotera. Za trećinu brži od CalComp 1023. Preko tri puta brži od HP DraftPro DXL. Drugim rečima, šta možete nacrtati za jedan sat na DMP ploteru, može trajati celo jutro na drugim.

KVALITET. Houston Instrument ploteri daju vrhunski kvalitet sa najboljom rezolucijom i ponovljivošću.

PRILAGODLJIVOST. Houston Instrument ploteri su jedini koji standardno omogućavaju crtanje bilo koje veličine crteža na bilo kojem formatu papira, pomoću svoje ekskluzivne Quick Scale™ funkcije.

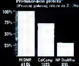
CENA. U svemu ovom, najbolje je to što Vam DMP-60 DL serija ploteru omogućava da Vi šijete svoju konkurenciju, po najkonkurentnijoj ceni.

Nazovite nas.



SCAN-CAD™ dodatak

Pretvorite Vaš DMP ploter u skener jednim stonim SCAN-CAD dodatkom. Skenirajte A0 crteže za delić cene velikog skenera.



HOUSTON INSTRUMENT™
A Summagraphics Company

Milentija Popovića 9
BEOGRAD (Sava centar)
Tel.. 011/222-43-22, lok. 615
Tel. 011/222-19-28
Fax : 011/455-785

* Columbia AI crtež, AutoCAD verzija 10 sa HP 7585 drajverom na COMPAQ 386-16 MHz računaru sa matematičkim koprocesorom. Ploteri su bili podešeni prema proizvođačkim uputstvima.

Ovlascene prodajna organizacija

OSA Beograd

SofWell d.o.o.
021/51-999, 52-396

EI „RAČUNARI" — Niš
Bul. V. Vlahovića 80
18000 Niš
Tel. 018/326-461, 55-583
fax 326-726

LANACO
78000 Banja Luka
Tel. 078/3 992, 48-717
fax 33-992

Makpetrol kompjuteri
fax 220-676



Za najzahtevnejše uporabnike urejevalnikov besedil je na voljo WORDSTAR LEGACY v okolju Windows,
za lastnike prenosnih računalnikov pa WORDSTAR LAPTOP COLLECTION.

Vse dodatne informacije dobite pri pooblaščenih prodajalcih Wordstarjevih proizvodov.

SRC d.o.o.
Tržaška 118 Tel.: 061/271-761
61111 Ljubljana Fax: 061/262-092

SRC
d.o.o. ljubljana

AUTODESK

AutoCAD R.11
AEC
AutoSHADE
AutoSKETCH
ANIMATOR
3D STUDIO

RASTEREX Graphics solutions

PROFESIONALNE
GRAFIČKE KARTICE:
LIGHT AT,LIBERTY
Autolcon
Rxlmage
SpotLight

Summagraphics

PLOTERI
GRAFIČKE TABLE
JetPro
DIGITAJZERI
SKENERI
SEKAČI

Genoa

OSNOVNE PLOČE:
386-33 MHz/64k,
486-33 MHz/256k
GRAFIČKE KARTICE

Chicony

TASTATURE

LAPTOP RAČUNARI



PROJEKTOVANJE SISTEMA
PC RAČUNARI SVIH KONFIGURACIJA
NOVELL MREZE
AutoCAD I POSLOVNE APLIKACIJE

RAČUNARSKI INŽENJERING
BEOGRAD

RAČUNARSKI INŽENJERING
COMPUTER ENGINEERING

Sedište/Head office
Mice Stojkovića 8,
11130 Beograd

Predstavništvo/Representative office
Šumadijski trg 6a,
11030 Beograd,

tel. 38 11 557-592
fax. 38 11 559-845

PROFESIONAL COMPUTER SYSTEMS **WEST** PREDUZEĆE „BSV" EXPORT IMPORT ELEKTRONIK CENTAR

AT 286-12/44

*80286-12 MHz CPU MAINBOARD DO 4 MB NA PLOČI O WAIT STATE
* HARD DISC IDE ST157A (44 MB)
*FLOPPY DISC 1,2MB 5.25"
*1×PARALEL 1×SERIAL PORT
*TASTATURA 102 TASTERA USA
*BABY KUĆIŠTE SA DISPLEJOM+200w
*14" MONOHROM MONITOR I KARTICA

CENA 1.799

AT 286-16/44

*80286 16MHz CPU MAINBOARD DO 4MB NA PLOČI O WAIT STATE
*HARD DISC IDE ST157A (44MB)
*FLOPPY DISC 1.2MB 5.25"
*1×PARALEL 1×SERIAL PORT
*TASTATURA 102 TASTERA USA
*BABY KUĆIŠTE SA DISPLEJOM+200w
*14" MONOHROM MONITOR I KARTICA

CENA 1.979

AT 386SX-16/44

*80386SX-16MHz CPU MAINBOARD DO 8MB NA PLOČI O WAIT STATE
*HARD DISC IDE ST157A (44MB)
*FLOPPY DISC 1.2MB 5.25"
*1×PARALEL 1×SERIAL PORT
*TASTATURA 102 TASTERA USA
*BABY KUĆIŠTE SA DISPLEJOM+200w
*14" MONOHROM

CENA 2.399

AT 386-20/44

*80386 20MHz CPU MAINBOARD DO 8MB NA PLOČI O WAIT STATE
*HARD DISC IDE ST157A (44MB)
*FLOPPY DISC 1.2MB 5.25
*1×PARALEL 1×SERIAL PORT
*TASTATURA 102 TASTERA USA
*BABY KUĆIŠTE SA DISPLEJOM+200w
*14" MONOHROM MONITOR I KARTICA

CENA 2.999

PORTABL TOSHIBA T 1000

*80C88-4.77 MHz CPU MAINBOARD 512K RAM
*3.5" 720K FLOPPY DISC
*SUPER TWIST LCD 80×25
KALENDAR+PARALEL+SERIAL
*MS DOS 2.11 U ROM-U
*2,9 Kg TEŽINA

CENA 1.682

OSNOVNE PLOČE

80286 12MHz SUNTAC DO 4MB	269
80286 16MHz NEAT DO 4MB	431
80386SX 16MHz DO 8MB	935
80386 20MHz DO8MB	1.430
80386 25MHzCACHE 32K DO 8MB	1.727
80386 33MHz CACHE 32K DO 8MB	2.418
80486 25MHz CACHE 8K DO 8MB	4.310
80486-25MHz CACHE 32K DO 8MB	5.399

KARTE ZA PROŠIRENJA

SERIAL KARTA	28
2×SER 1×PAR 1×GAME PORT	48
2×HDC 2×FDC MFM KONTROLER 1:1	143
AT BUS (IDE) 2×HDC 2×FDC	49
AT BUS 2×HDC 2×FDC 1×SER 1×PAR	85

PREDUZEĆE BSV
Pionirska 5
31300 Prijepolje
tel/fax **033/21-258**
Žiro račun 63000-601-1877

VIDEO KARTE

HERCULES (720×348)+PAR	42
EGA (640×480)	173
VGA (800×600)	208
VGA 16BIT 512K (1024×768)	286

FLOPI DISKOVI

1.2MB 5,25"	169
1.44MB 3.5"	168

HARD DISKOVI

ST 44MB IDE 28ms	496
ST 89MB IDE 19ms	872
ST 125MB IDE 19ms	1.128
ST 204MB IDE 15ms	1.757
ST 338MB IDE 16ms	3.283

MONITORI

14"MONOHROM	244
14" VGA MONOHROM	440
14" VGA COLOR 1024×768	890
14" MULTISYNC 2A SSI 800×600	1.606

TASTATURA

102 TASTERA AMERIČKA	88
TASTATURA SA KLIKOM	136

ŠTAMPAČI

EPSON LX400 9PIN A4	625
EPSON Q400 24PIN A4	879
AMSTRAD Q5000 24PIN A3	1.199
HP LASERJET IIP	3.339

PLOTER

HP 7570 A1	9.999

MIŠEVI

BEŽIČNI MIŠ	173
GENIUS GM-6000 MS. COMP	86

DISKETE

5,25" 1.2MB 10kom	17
3,5" 1.44MB 10kom	25

KOPROCESORI

80287 10MHz	360	80387 20MHz	863
80287 12MHz	447	80387 25MHz	1.151
80387SX 16MHz	792	80387 33MHz	1.439
80387 16MHZ	810		

RAMOVI

41256-100	8
411000-80	18
SIMM 1MB 70ns	159
SIPP 1MB 70s	166

SVE NAVEDENE CENE SU U DEM
SA URAČUNATOM CARINOM I POREZOM.
ISPORUKA ODMAH. GARANCIJA GODINU DANA
OBEZBEĐEN SERVIS NA CELOJ TERITORIJI SFRJ.

MZ
Preduzeće „MZ" 11000 Beograd
Dubljanska br. 70 Tel. 011/434-812,
Fax 011 450 471

ŠTAMPAČI

EPSON

LX-400 9-iglični A4 format 180 znakova/s
LX-850 9-iglični A4 format 200 znakova/s
FX 850m 9 iglični A4 format 300 znakova/s
LX-1050 9-iglični A3 format 200 znakova/s
FX-1050 9-iglični, A3 format 300 znakova/s
LQ-550 24-iglični A4 format 180 znakova/s
LQ-850+ 24-iglični A4 format 300 znakova/s
LQ 1010 24-iglični A3 format 180 znakova/s
LQ-1050 +24 iglični A3 format 300 znakova/s
LQ-860 24-iglični A4 format kolor 300 znakova/s
LQ 1060 24 iglični A3 format kolor 300 znakova/s
LQ-2550 24 iglični A3 format kolor 400 znakova/s
DFX 5000 »heavy duty« 9 iglični 533 znakova/s A3 format
DFX-8000 »heavy duty« 9-iglični A3 format 1066 znakova/s
EPL 7100 laserski štampač MEGA II 2 Mb 6 str/min
GT 4000 color scanner
GT 6000 color scanner
GT 1000 handy scanner

Roland DIGITAL GROUP

DXY 1100 A3 PLOTER
DXY 1200 A3 PLOTER
DXY 1300 A3 PLOTER
DPX 2500 A2 PLOTER
DPX 3500 A1 PLOTER
GRX 300 A1 „ROLL" PLOTER
GRX-400 A0 „ROLL" PLOTER
DPX-4600 A0 PLOTER
LTX 100 A3 TERMIČKI PLOTER
LTX 120 A3 TERMIČKI PLOTER
LTX 321 A1 TERMIČKI PLOTER
LTX 420 A0 TERMIČKI PLOTER
CAMM-1 PNC 1000 PLOTER REZAČ
CAMM-1 PNC 1800 PLOTER REZAČ A0 FORMAT
STIKA REZAČ SA UGRAĐENIM SKENEROM
SKETCHMATE A4 PLOTER
ROLAND PERA ZA PLOTERE

EIZO®

MONITOR EIZO 4050
MONITOR EIZO 9060S
MONITOR EIZO 9070S
MONITOR EIZO 9400I
GRAFIČKA KARTICA EIZO MD B10
GRAFIČKA KARTICA MD B12-60

ISPORUKA ODMAH
GARANCIJA GODINU DANA
POSEDUJEMO DODATNI
I POTROŠNI MATERIJAL
INFORMACIJE NA TEL. 011 434 812
FAX 011 450 471


EPSON
VELIKO IME, POUZDAN KVALITET!
Avtotehna Titova 36 i Celovška 228 Ljubljana
Mladinska knjiga Veletrgovina Titova 145 Ljubljana
Mladinska knjiga Trgovina Titova 3 Ljubljana
Gambit p o Titova 8 Ljubljana
ATR Ljubljana Birostroj
Mogućnost leasing prodaje!
EPSON
EPSON
d o o
CELOVŠKA 175 • YU
61107 LJUBLJANA
R E P R O
L J U B L J A N A
TELEFON 061/552-341 552-150, 554-450 FAX (061) 552-563
TLX 31 639 yu-antena p p. 69
POUZDANO IME, VELIKI KVALITET!

METALOPRERADA — BAČ

21420 Bač, Ul. JNA 137
Telefoni: (021) direktor 770-313, komerc 770-435,
centrala 770-520, 770-524, 770-539, faks 771-066

Ako želite da vaša kancelarija ili radna soba bude savršena

RADNI STOLOVI ZA PERSONALNE RAČUNARE



Namena

- OPREMANJE POSLOVNIH PROSTORA ČIJI SADRŽAJ ČINE PERSONALNI RAČUNARI
- OPREMANJE PRIVATNIH PROSTORIJA U KOJIMA SU PRISUTNI PERSONALNI RAČUNARI
- OPREMANJE ENTERIJERA U KOJIMA SE ŽELI RACIONALISATI SA PROSTOROM I OBEZBEDITI BESPREKORNA FUNKCIONALNOST RAČUNARSKE OPREME (računar, tastatura, štampač, UPS, kutije za diskete), KAO I VIDEA, TELEVIZORA, MUZIČKOG STUBA

Osobine

- BESPREKORNA FUNKCIONALNOST
- JEDNOSTAVNA KONSTRUKCIJA
- POKRETLJIVOST
- PRILAGODLJIVOST ENTERIJERU
- PRILAGODLJIVOST UZRASTU I VISINI KORISNIKA
- SMANJENJE ZAMORA, ZBOG MOGUĆNOSTI PODEŠAVANJA PREMA FIZIČKOJ KONSTRUKCIJI KORISNIKA
- VELIKA LEPEZA BOJA
- UŠTEDA U PROSTORU

Dimenzije

- DUŽINA 600 mm
- ŠIRINA 600 mm
- VISINA
min 980 mm,
max 1300 mm











MOST JUNIOR

Procesor 80286 16Mhz, 1Mb RAM–0WS, Landmark 21 IDE tvrdi disk 40Mb »Conner« CP3000/3044, floppy 5 25"/1 2Mb ili 3,5"/1 44Mb, AT BUS adapter sa 2 serijska i jednim paralelnim portom, »Hercules« grafička kartica B/W monitor »Sanyo«, desktop kućište 200W i tastatura »Cherry«–YU/ASCII

CENA. 30 000

MOST BABY

Procesor 80286 12Mhz, 1Mb RAM–0WS, Landmark 16, IDE tvrdi disk 40Mb »Conner« CP3000/3044 floppy 5 25"/1 2Mb ili 3 5"/1 44Mb AT BUS adapter sa 2 serijska i jednim paralelnim portom, »Hercules« grafička kartica B/W monitor »Sanyo« desktop kućište 200W i tastatura »Cherry« YU/ASCII

CENA 28.500



MOST ROSE

Procesor 80386SX 20Mhz, 2Mb RAM, IDE tvrdi disk 45Mb »Seagate« ST157A, floppy 5.25"/1,2Mb ili 3,5"/1,44Mb, AT-BUS adapter sa 2 serijska i jednim paralelnim portom, »Hercules« grafička kartica, B/W monitor »Sanyo«, mini tower kućište 200W i tastatura »Cherry«–YU/ASCII.

CENA. 39 000



MOST STAR

Procesor 80486-25Mhz, 128kB cache memorija, 8Mb RAM, IDE tvrdi disk 125Mb »Seagate« ST1144 floppy 5 25"/1,2Mb ili 3,5"/1,44Mb, AT BUS adapter sa 2 serijska i jednim paralelnim portom, »Hercules« grafička kartica, B/W monitor »Sanyo«, tower kućište 230W i tastatura »Cherry«–YU/ASCII.

CENA. 119 000

MOST EXPERT

Procesor 80386 25Mhz, 64kB cache memorija, 4Mb RAM, IDE tvrdi disk 40Mb »Conner« CP3000/3044 floppy 5 25"/1 2Mb ili 3,5"/1,44Mb, AT BUS adapter sa 2 serijska i jednim paralelnim portom, »Hercules« grafička kartica B/W monitor »Sanyo«, tower kućište 230W i tastatura »Cherry«–YU/ASCII

CENA. 62.000



MOST ASSISTANT

Procesor 80386 33Mhz, 64kB cache memorija 4Mb RAM IDE tvrdi disk 40Mb »Conner« CP3000/3044, floppy 5 25"/1,2Mb ili 3 5"/1,44Mb AT BUS adapter sa 2 serijska i jednim paralelnim portom »Hercules« grafička kartica, B/W monitor »Sanyo« tower kućište 230W i tastatura »Cherry«–YU/ASCII.

CENA. 72.000

DOPLATE NA OSNOVNE KONFIGURACIJE

1Mb RAM	2.500
Floppy disk »Teac« 3.5", 1 44Mb	2.850
VGA monitor »Tystar« i Super VGA graf.adapter 1024x768/1Mb	15 000
Tvrdi disk »Seagate« 1144A 125Mb IDE	13.000
Tvrdi disk »Conner« CP3100, 100Mb SCSI	31 000
Tvrdi disk »Conner« CP3204 209Mb IDE	31 000
Mini-tower kućište 200W	2.000
Tower kućište 230W	3 000
Koprocesor 287/10MHz	7.500
Koprocesor 287/12MHz	12.000
Koprocesor 387/25MHz	20.000
Koprocesor 387/33MHz	23.000

MIŠEVI I SKENERI, MODEMI

Miš »A4 Tech« sa mouse padom i softverom	1.850
Handy scanner »A4 Scan« sa OCR softverom	7 300
Page scanner »Microtec« 300Z, A4, color i gray scale	73.000
Modem »Dynalink«, 2400 bauda interni	6 000

ŠTAMPAČI

»EPSON« LX-400 A4, 9–pinski, YU set	10 000
»EPSON« LQ 1010, A3, 24–pinski 180 cps, YU set	28 500
HP Laser Jet IIP 512kb	51 000
HP Laser Jet III 1Mb	79 500
HP LaserJet IIIP!	Zovite!
»Pacific 4 Memory«, Pacific Data Products, mem proširenje 2MB za HP LJ IIP i LJ III	14 000

TVRDI DISKOVI

»Seagate« ST157A, 44MB, IDE, 28ms	9 750
»Seagate« ST1144A, 125Mb IDE	18 000
»Conner« CP3044/3000 40Mb IDE	10 500
»Conner« CP3204 209Mb IDE	36 000
»Conner« CP3100 100Mb SCSI	23.000
»IBM« 320Mb 12ms, SCSI	65 600
»Adaptec« AHA-1542B, SCSI kontroler, podržava 8 SCSI uređaja	17 000
IDE adapter 2HDD/2FDD/2S/1P/1G	2.300

MONITORI I GRAFIČKE KARTICE

»Sanyo«, 14" »Hercules« kompatibilan	6.000
»Tystar«, VGA 1024x768, 14" 028" dp	16.200
»EIZO« Flexscan 9070s, VGA 1024x768, 16" multisync	49 000
»SONY« CPD 1402E/5 Multiscan, VGA 1024x768, 14" 0 25" dp, multisync	24 000
»Hitachi« CM 2085MU VGA 1280x1024, 20" multisync	81 250
»Magnavox« mono VGA 800x600, 14" (»Philips« američka varijanta)	Zovite!
»Hercules« kompatibilni grafički adapter	1.500
VGA adapter 1024x768/1Mb, »Tseng Labs« ET4000 »Trident« TVGA8900, drajveri za Windows 3 0, AutoCAD 10, Word Perfect 5 1 ..	8.000

RAČUNARSKE MREŽE

»Novell« NetWare v2.2, 5 korisnika	26.000
»Novell« NetWare v2.2, 10 korisnika	57 500
»Novell« NetWare v2.2, 50 korisnika	101 000
»Novell« NetWare v3.11, 20 korisnika	102.000
»Novell« NetWare v3 11, 100 korisnika	202.000
»Novell« NetWare Requester for OS/2, v1 2	1.500
Ethernet adapter 8–bitni (NE-1000 ili 3C501 kompatibilan)	8.000
Ethernet adapter 16–bitni (NE-2000 komp	10.500

SOFTVER

»Microsoft« MS-DOS 4.01	3.200
»Microsoft« Windows 3 0	4.000
»XEROX« Ventura 3 0 (GEM i Windows)	24.300
»Corel« CorelDRAW! 2.0	17 000
»Borland« Turbo C++ 2.0	7 000
»Borland« Turbo C++ 2.0 Professional	8.750
Fontovi za CorelDRAW/PostScript/Laser Jet	
YU latinice za CorelDRAW/PostScript	1 500
Ćirilice za CorelDRAW/PostScript	2.700

Posebno povoljni uslovi za dalju prodaju Popusi na količine (5-9, 10-19, 20 i više) Obratite nam se za poseban dealerski cenovnik, kao i za uslove dealerske prodaje! Cene su podložne promeni, u zavisnosti od promena kursa dinara Uplate virmanom ili uplatnicom na žiro račun 60802-601 66178 SDK Beograd

Preduzeće »Most«, tel 422–170, fax/tel. 426–144 Beograd, Kralja Bodina 2, radno vreme 9-17. Stalna prezentacija opreme u Sava Centru, galerija Inter art,

MOST

MI SE RAZLIKUJEMO OD DRUGIH. NAS MOŽETE DOBITI ODMAH.

ELITE & EUROHIT Computers

BEOGRAD, Vodovodska 80 Tel/Fax: (011) 516-253

MI SMO SIGURNI DA SU KVALITET I DOBRA CENA USLOV ZA OPSTANAK!

Naša ponuda:
STANDARDNA KONFIGURACIJA:
- **HD 44,6 MB**
- **1 MB RAM**
- **FLOPI DISK 1,2 MB**
- **TASTATURA KLIK,**
- **KUĆIŠTE BABY, PS 200 W**
- **2P/2S**

PC AT 286/16 MHz, Mono	1899 DEM
PC AT 286/16 MHz, VGA Monitor VGA kolor 1024x768, VGA grafička karta, 16 bit-a, 512 KB	2599 DEM
PC 386Sx 16/20 MHz	PITATI
PC 386/33 MHz sa 64 KB CACHE sa 4 MB RAMA	3700 DEM

Panasonic

PRINTERI PANASONIK - 24 pina

A4...240 cps	970 DEM
A3 ..240 cps	1250 DEM
Miševi	65 DEM
Modemi interni 2400 BPS	270 DEM
Saund Blaster	Pozvati

Radno vreme: od 9°° - 17°°

Servis i garancija obezbeđeni

Kod nas prvo proverite robu, pa tek onda placate u dinarskoj protivvrednosti.

ISPORUKA ODMAH - SVE CENE SU ZA KRAJNJEG KUPCA

Micro Computing

41040 ZAGREB, Fočanska 35
tel. 041/259-686 (8-20 sati) i 041/511-139 (17-21 sat)
fax 041/259-686

ATARI

AT SPEED C16 AT SPEED, PC SPEED
Hard diskovi (40Mb do 170Mb) cache 12/19 ms
MEGAFILE 20 30 i 60
MEGAFILE 44 izmjenjiv disk idealno za DTP
Ugradnja Quantum diskova u MEGA ST računare
Flopy disk 3.5 i 5 25
Pregradnja SF 354 u dvostrani disk
Hypercache 16 Mhz/32kb cache/ 100% brb ST
Handy scanner 400 dpi, 32 nijanse, software
Modem fax (1200-9600) sa softwer om
RAM ST - 1Mb 2.5Mb 4Mb STE - 2Mb i 4Mb
HF modulator i videodigitalizator
Laser interface, TOS 1 4
Monitor SM 124
Scart, kompozitni i centroniks kablovi
Najnoviji opto mehanički SUPER-MIŠ (290 dpi)
SERVIS 260 520 1040 i Mega ST računara

AMIGA 500 i 2000

- ATONCE, AT emulator
- Hard diskovi od 20Mb do 105Mb
- TURBO kartice 14Mhz
- GENLOCK (PAL, Y-C i PROFI)
- Floppy diskovi 3,5" i 5 25"
- Soundsampler i videodigitalizator
- Proširenje memorije od 512Kb do 2Mb
- Modem-fax (1200-9600) sa software om
- Najnoviji SUPER MIŠ (290 dpi)
- MIDI interface
- HF modulator i SCART kabel
- KICKSTART 1 3 i BIG AGNUS 8372A
- SERVIS A 500 i A 2000
- FILECARD (40Mb 170Mb)
- RAMCARD (2Mb - 8Mb)

PONUDA MJESECA

1. Najnoviji računari MEGA STE2 ili MEGA STE4 sa SM 124. 16Mhz, Copro., Cache, HD 48Mb/28ms. TT-TOS 2.0 i TT-kućište
2. MEGA ST2 ili MEGA ST4 sa SM 124
3. SUNSHINE-PAKET 1040 STE + SM 124 + Joystick + POWER PACK (20 igara) + originalni program ADIMENS i THAT'S WRITE + ATARI torba
4. AT SPEED C16 16Mhz, Coprocessor, EGA i VGA NORTON 6.5 Dr DOS 5.0
5. ATONCE AT emulator za AMIGU 500 i 2000: Procesor 80286/8, NORTON 6 2 CGA Hercules, oliveti, toshiba, EGA i VGA kartice, podržani svi portovi te floppy diskovi
6. SUPER MIŠ najnoviji opto-mehanički miš sa 290 dpi rezolucijom, Microsoft-ov ergonomični design, mikro prekidači, precizne i tiha kugica, koristi 70% manje prostora, univerzalan za ST STE, TT i AMIGU.


WordPerfect® familija
WordPerfect
vodeći tekst procesor na svjetskom tržištu
preko 4,000 000 korisnika
jedinstvena sprega snage i jednostavnosti
preko 500 podržanih printera
tabele, matematika, jednadžbe
prethodni pregled dokumenta
integracija grafike u tekst
DataPerfect
relacijska baza podataka
jednostavno generiranje ekrana
jednostavno generiranje izvještaja
fleksibilni pregled podataka
run-time modul
PlanPerfect
moćan tabelarni kalkulator
izvanredne grafičke mogućnosti
optimizacija preračunavanja
preko 2.000.000 celija po tabeli
preko 100 specijaliziranih funkcija
upotreba virtuelne memorije
povezivanje tabela
WordPerfect OFFICE
integracija svih WP produkata
potpuno rješenje automatizacije ureda
pristup do drugih aplikacija
elektronička pošta (e-mail)
grupni terminski kalendar
WAN i globalna komunikacija
kalkulator, rokovnik
DrawPerfect
vrhunski paket za poslovnu grafiku
preko 500 predefiniranih slika
tabele, grafikoni
vektorska definicija objekata
ulaz podataka iz 14 grafičkih formata
izlaz na jedinice visoke rezolucije
30 matičnih fontova
run-time modul
IQ ENGINEERING
Super Cartridge 1IE
YU slova iz WordPerfect-a na HP laseru
19 fontova
vrlo mali fontovi (4-6 t.) za spreadsheet
veličine od 4-16 točaka
Portrait i Landscape orjentacija
SAMO NAJBOLJE JE DOVOLJNO DOBRO!
perpetuum d.o.o.
integralni informacijski sistemi
NOVA ADRESA: Kučerina 5, 41000 Zagreb, tel. 317-020, 328-999, fax:328-999

Wearnes
BOLDLINE M SERIES

Za većinu računala na tržištu, karakteristično je da im vrijednost počinje rapidno opadati od trenutka kupnje.

Dakle, ne baš povoljna perspektiva, uzmemo li u obzir uložena sredstva.



No u slučaju Wearnes Boldline M serije situacija je bitno drugačija.

Naime ova računala praktički nikad ne zastarijevaju, pa nema potrebe da ih zamjenjujemo novim.

Osnovni model Wearnes 286 računala može se bez problema pregraditi u 386SX mašinu ili štoviše u i486 stroj.

I sve to u roku od nekoliko minuta.

Zahvaljujući specijalnom adapteru na matičnoj ploči računala.

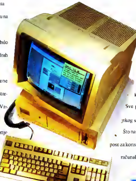
Dakle bez potrebe da se koristi bilo koji od ukupno 6 raspoloživih slobodnih priključaka za proširenja.

Osim svojevrsne ugrađene "budućnosti", umjesto uobičajenog "zastarijevanja", u našim računalima očekuju Vas i druga iznenađenja.

Brzi odziv ugrađene radne memorije kapaciteta 1 MB bez statusa čekanja.

Kruti disk kapaciteta 40 MB.

B*oldline M Serija. Kraj ere potrošnih računala.*



Kombinacija: IBM i ALR kompatibilnosti.

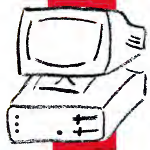
Sve po cijenama nižim od industrijskog standarda.

Što nas čini pouzdanom opcijom, naspose za korisnike koji ne vjeruju da novac i računalo mogu zastarjeti.

ORIA
61410 Zagorje ob Savi
Polje 4
tel. +38(0)601/61-477
61 111, 61-149, 61-235
fax (0601) 61-175

Wearnes
Computer Systems

ComTrad *the art of computer making*



286 AT

- ✓ INTEL286 CPU, 12 MHZ
- ✓ 1 MBYTE RAM, EXP. UP TO 4 MBYTE
- ✓ VGA DISPLAY CARD
- ✓ 1 DISC DRIVE 5.25", 1.2 MBYTE
- ✓ 1 DISC DRIVE 3.5", 1.44 MBYTE
- ✓ 43 MBYTE IDE-BUS HARD DISC
- ✓ 2 SERIAL PORTS
- ✓ 1 PARALLEL PORT
- ✓ 1 GAME PORT
- ✓ SERIAL MOUSE
- ✓ VGA COLOR MONITOR
- ✓ KEYBOARD WITH 101 KEYS

386SX with the same configuration **2.499.- DM**

1.997.- DM

ComTrad
HIGH SPEED PC'S

ComTrad
CT Computer Trading
Wohlerstraße 12
4000 Dusseldorf 30
Germany
Tel. 9949-211-6508-0
Fax. 9949-211-650840

ComTrad Information
Vladimira Popovića 6
11000 Beograd
Tel. 011-2224-151
Tel. 011-2222-652
Tel. 011-2224-433/8265
Fax. 011-2223-951

KMA D.O.O.
Milentija Popovića 9
11000 Beograd (Sava Centar)
Tel. 011-222-4522 lok. 615
Tel. 011-222-1928
Fax. 011-455-785

ComTrad
Kranjceva 4
41000 Zag
Tel. 041 21
Fax. 041 21

SCO

THE SANTA CRUZ OPERATION

Softverska kompanija 1990. godine

Operativna okruženja

SCO System V

Porodica SCO Sistema V obuhvata SCO UNIX System V/386 Release 3 2 i SCo XENIX System V. višekorisnički, multitasking operativni sistemi za mikroračunare bazirani na industrijskim standardima

SCO UIX System V/386 Release 3.2

Robustna, potpuno licencna verzija AT&T UNIX Sistema V sa dodatnim poboljšanjima koja je isključivo ugradio SCO

SCO XENIX

Industrijski standardan UNIX sistem za personalne računare bazirane na intelu

SCO MPX

Multiprosecsorsko proširenje za računare bazirane na 386 i 486 koji rade pod SCO UNIX Systemom V/386

SCO VP/ix

Pod SCO VP/ix, višestruke MS-DOS aplikacije mogu ići paralelno sa konvencionalnim XENIX i UNIX sistemima.

Komunikacije i mreže

SCO TCP/IP

Povezivanje preko mreže između raznih operativnih sistema i računara

SCO NFS

Networked File i Data Sharing za SCO UNIX i Open Desktop

SCO XENIX-NET

Microsoft mreže za XENIX sisteme. Lokalna mreža namenjena da specifično poveže personalne računare koji rade pod XENIX-om i DOS-om

SCO UniPATH SNA-3270

Višekorisnička veza između mikro i glavnog računara — ulaz viškorisničkih personalaca u IBM SNA mreže

Radne stanice

Open Deskop

Potpun grafički operativni sistem. Open Desktop vam omogućava da koristite moćne grafičke aplikacije u prozorima, da manipulišete podacima, pristupate datotekama i aplikacijama na umreženim sistemima i da koristite DOS i UNIX sisteme

Poslovna automatizacija

Microsoft Word 5.0

Višekorisnički tekst procesor za UNIX sisteme

SCO Portfolio

User-friendly interfejs za SCO System V uključujući aparate za automatizaciju kancelarijskog poslovanja kao što su elektronska pošta (e-mail) i kalendar

SCO Professional

Lotus 1-2-3 workalike za SCO System V

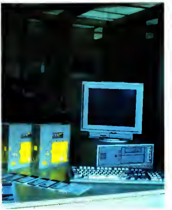
SCO FoxBASE+

dBASE III plus workalike za SCO System V

Svetski lider u rešenjima za softver otvorenih sistema

MAC RITA AWARDS PC COMPUTING

SCO je dobio priznanje kao Softverska Kompanija Godine na ceremoniji dodele RITA nagrada



DISTRIBUTER ZA JUGOSLAVIJU

ENERGOPROJEKT

HOLDING KORPORACIJA

MDD ENERGODATA

11070 Beograd, Lenjinov bulevar 12, p. fah 20
Tel (011) 138-520, 139-712, 222-3072
Fah (011) 222-4780 • Tlx ENDATA YU 11764



HVALA BORLAND

od 2. aprila dalje



Preduzeće MARAND generalni zastupnik BORLANDa za Jugoslaviju - vam u vremenu trajanja akcije [illegible] BORLANDova programske pakete:
PARADOX 3.5, PARADOX ENGINE 2.0, QUATTRO PRO 2.0, TURBO PASCAL 6.0, BORLAND C++ 2.0, SIDEKICK 2.0

NAROČITO POVOLJNO !

Prva povoljnost: svoje piratske programe možete legalizovati sa originalnim BORLANDovim programima

Veća povoljnost: primit ćete poslednju verziju željenog programa bez obzira na to koju verziju primenjujete

Još veća povoljnost: programske pakete sa kompletnom odgovarajućom literaturom moći ćete za vreme akcije nabaviti 50 % jeftinije

Najveća povoljnost: kupnjom bilo kojeg BORLANDovog programskog paketa ulazite u čarobni svet najboljih svetskih PC programa što će vam omogućiti stalno nadgrađivanje (upgrade) uz osetno snižene cene

Naročita povoljnost: kao partneri BORLANDa redovito ćete primati sva obaveštenja o najnovijim dostignućima na području računarskog softvera

Povoljnost da se...: nakon završene akcije mi ćemo među onima koji će nam poslati popunjene kupone i koji će se u tom vremenu uključiti u sistem BORLAND ždrebom izabrati 10 dobitnika nagrada

1. nagrada: PC AT RAČUNAR MARAND F-16/3

P.S. Povoljnost: bez obzira na to koji BORLANDov program primenjujete (odn. konkurentni program bilo kojeg proizvođača) ili ako ste registrovani u nekoj stranoj zemlji bit ćete u mogućnosti da svoje programe nadgradite odgovarajućim najnovijim BORLANDovim programskim paketom

DISTRIBUTERI

(061) 318 736 625
211 047 MAOP
221 030 MEDIIA d o o
329 244 MIKRODATA
347 361 MK KIKOOPREMA
211 095 MK KNJIGARNA
557 790 QUANTUM d o o
318 660 IRIAS
(068) 22 000 ALAB d o o
25 999 FENIX d o o
(041) 418 582 ABL
433 722 MK KNJIGARNA
(011) 603 390 CET
40K 5472 REY d o o
(021) 52 396 SOFTWELL d o o
(024) 851 532 AMIKROSOFTAGENC d o o



OMOGUĆUJE

MARAND d o o
Generalni zastupnik
BORLAND-a za Jugoslaviju
Kardeljeva ploščad 24
61000 LJUBLJANA
Tel. 061 340 652, 371 114
Fax 061 342 757

MARAND

Molimo vas da nam pošaljete dodatne informacije o programskim paketima.

a	PARADOX 3 5	Ime.	
b	PARADOX ENGINE 2.0	Prezime.	
c.	QUATTRO PRO 2 0	Adresa.	
d	TURBO PASCAL 6 0		
e.	BORLAND C++ 2.0	Preduzeće	
f.	SIDEKICK 2 0	Telefon	

MCH group computer-systeme, computer, solution, tehnologies

MCH Computer d.o.o.
62000 Maribor Tomsičeva 19
Tel. & Fax (062) 28 250

&

MCH Solution d.o.o.
11000 Beograd Zaplanjska 86 Tel (011) 468 732
Fax (011) 467 059

&

MCH Tehnologies d.o.o.
41000 Zagreb Proleterskih brigad 78 Tel. (041) 539 892
Fax (041) 538 946

COMPUTER SHOP
A-8472 STRASS/Stmk Hofgrath 2 AUSTRIA
Tel 9943 34 53 44 50 Fax 9943 34 53 43 65

Poštovani kupci !

Nasa trgovina sa računarima i pripadajućim komponentama nalazi se samo 3 km od graničnog prelaza Šentilj u smeru Leibnitz-a.
Povoljne cene jednogodišnja garancije i sevis u Mariboru !
U cenu je uracunato i sastavljanje računara u Mariboru !
Garancije 12 mesecev !

Dodite i uverite se ili nas nazovite na naš telefon !
Govorimo slovenski i srpskohrvatski !

Zastupamo

* PEACOCK
* WESTERN DIGITAL
* SIGMA DESIGNS
* TALLGRASS
* MITSUBISHI
* SEAGATE
* INTEL

ZA VEČE KOLIČINE NUDIMO VRLO POVOLJNE RABATE !
POSEBAN POPUST ZA TASTATURE CHERRY - YU !

AT286/12

AT286/12 1MB RAM
Hercules komp graf kartica
Baby Tower Case 200 W
42 Mb/25 ms HDD West Digital
AT-BUS Kontroler 1:1
1,2 MB 5.25. TEAC FDD
MF Tastatura 102-YU CHERRY
14, Monitor Mono SAMSUNG

CENA 1.483,- DEM Netto

AT286/12 VGA

AT286/12 1MB RAM
VGA Graf. Kartica 256 Kb 16 bit
Baby Tower Case 200 W
42 Mb/25 ms HDD West Digital
AT BUS Kontroler 1:1
1.2 MB 5.25 TEAC FDD
MF Tastatura 102-YU CHERRY
14, VGA Monitor Mono

CENA 1.815,- DEM Netto

FAX
(062) 28 250
(011) 467 059

Želite ponudu ? Nema problema !
Ispunite i pošaljite nam na naš fax !
Odgovorićemo ODMAH !

FAX
(041) 538 946
9943 34 53 4365

Gosp.			*Firma*		
Ulica			*Mesto*		
Tel.			*Fax*		
Kućište	Procesor	Koprocesor	Tvrdi disk	Monitor	Streamer
☐ Desktop	☐ 80286/12	☐ 80x87	☐ 40 MB	☐ Mono 14	☐ 40 MB
	☐ 80386SX/16		☐ 80 MB	☐ VGA mono 14.	☐ 80 MB
			☐ 111 MB	☐ VGA color 14.	☐ 150 MB
☐ Mini tower	☐ 80386SX/16	RAM	☐ 120 MB	☐ VGA color 16.	
	☐ 80386/25	☐ 1 MB	☐ 160 MB	☐ 20 C/B	Operativni sistem
	☐ 80386/25C	☐ 2 MB	☐ 336 MB	(1660 x 1200)	☐ MS DOS 3.3
	☐ 80386/33C	☐ 4 MB	☐ 676 MB	☐ 20 Multisync	☐ MS DOS 4.01
		☐ 8 MB	☐ više od	color	☐ Unix
☐ Tower	☐ 80386/25C	☐ 16 MB	676 MB		☐ Xenix
	☐ 80386/33C	☐ više od 16 MB	in to	Flopy disk	
		i to		☐ 3.5. 1.44 MB	
☐ Laptop	☐ 80286/12			☐ 5.25. 1.2 MB	

☐ Pošaljite cjelokupan cenovnik — ☐ Štampaci (EPSON) ____
☐ Zanimaju me mreže (NOVELL, ...) ____
☐ Pošaljite nam što više informacije ! — ☐ Interesuje me vaš servis ? — ☐ Pošaljite prospekte !

MCH computer systeme

AUTRONIC

MCH Computer-Systeme
Handelsgesellschaft m b H
8472 Strass/Stmk, Hofgreith 2
Tel 9943 34 53 44 50
Fax 9943 34 53 43 65

PEACOCK COMPUTER

AUTRONIC Computer Systeme
A 9020 Klagenfurt Radetzkystr 18
Tel 9943 463 51 48 71,
Fax. 9943 463 51 48 73

Osnovne konfiguracije

AT 286 - 16	1 MB RAM, VGA , 40 MB tvrdi disk DOS 4 01	47 821,- DIN
AT 286 - 16	1 MB RAM, VGA , 110 MB tvrdi disk, DOS 4 01	63,279,- DIN
AT 386 - 25	2 MB RAM, VGA , 40 MB tvrdi disk DOS 4 01, WINDOWS 3 0 + MS Mouse	81 282 - DIN
AT 386 - 25	2 MB RAM, VGA, 110 MB tvrdi disk, DOS 4 01, WINDOWS 3 0 + MS Mouse	96,740,- DIN
AT 386 - 25C	4 MB RAM, VGA, 110 MB tvrdi disk, DOS 4 01, WINDOWS 3 0 + MS Mouse	114,153,- DIN
AT 386 - 33C	4 MB RAM, VGA, 110 MB tvrdi disk DOS 4 01, WINDOWS 3 0 + MS Mouse	127 159 - DIN
AT 486 - 25C	4 MB RAM, VGA, 110 MB tvrdi disk, DOS 4 01, WINDOWS 3 0 + MS Mouse	163 259,- DIN
MONITOR	VGA MONO MONITOR 640 x480	5,941,- DIN
MONITOR	VGA TRISCAN MONITOR 1024 x 768	22,843,- DIN

Distributeri :

MCH Computer d.o.o.
62000 Maribor, Tomšičeva 19, Tel & Fax (062) 28 250
MCH Solution d.o.o.
11000 Beograd, Zaplanjska 86, Tel (011) 468 732,
Fax (011) 467 059
MCH Tehnologies d.o.o.
41000 Zagreb, Proleterskih brigad 78, Tel (041) 539 892,
Fax (041) 538 946

AUTRONIC d.o.o.
61000 Ljubljana, Kardeljeva ploščad 17
Tel (061) 345 161
Tel & Fax (061) 302 581
AUTRONIC d.o.o.
41000 Zagreb, Kollerova 3
Tel & Fax (041) 232 259





Microsoft

Povoljan vetar u jedrima!

Od davne 1976 godine, Microsoft znači povoljan vetar u jedrima za sve putnike na našem brodu.

Pružamo stabilnu platformu operacionih sistema, od MS-DOS preko MS Windows do OS/2. Microsoft-DOS pokreće 60 miliona PC računara širom sveta. Windows 3.0 ih čini upotrebljivim, a snažni OS/2 i lokalna mreža Lan Manager 2.0 ih spajaju.

Brojne softverske kuće, naši saputnici, su na našim operacionim sistemima zaradile dobro ime i ostvarile sjajne produkte. Njihovi programeri su se često služili našim programskim jezicima. U stvari, Microsoft C ili Macro Assembler su oruđa, pomoću kojih su izrađeni brojni popularni programi te ih upotrebljava većina profesionalnih programera.

Osim C 6.0 i MASM, kod Microsofta ćete naći i standardne jezike velikih sistema. Cobol, Fortran, Basic i Pascal.

Uz operacione sisteme i programske jezike, Microsoft proizvodi aplikacije koje su postale standardom. Word među tekst procesorima, Project među programima za vođenje projekata, Excel među radnim tabelama... ili Works, integrirani paket za ličnu upotrebu po povoljnoj ceni. Već sada su Microsoftove aplikacije na raspolaganju za sve platforme i u svakom momentu predstavljaju najviši domet tehnologije i pružaju najveće mogućnosti, jer MS ulaže u razvoj više nego njegovi glavni konkurenti zajedno.

Microsoft programsku opremu u Jugoslaviji distribuiše **Atlantis**, poslovni informacijski sistemi, d.o.o. Rado ćemo odgovoriti na vaša pitanja u vezi MS produkata i cena i uputiti vas na ovlaščenog Microsoft dilera koji prodaje MS softver sa važećom registracijom i pravom na nove verzije. Školskim ustanovama nudimo mogućnost kupovine MS softvera po niskim nekomercijalnim cenama.

ATLANTIS d.o.o.
POSLOVNI
INFORMACIJSKI
SISTEMI
Cankarjeva 4
61000 Ljubljana
tel/fax
(061) 221 608

ATLANTIS

BLD

Microsoft je zaštićni znak Microsoft, Inc.